AIMS Mathematics, 11(6): 15926–15951.
DOI: 10.3934/math.2026656
Received: 17 April 2026
Revised: 19 May 2026
Accepted: 26 May 2026
Published: 05 June 2026

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

Research article

Numerical investigation of the time-fractional Fisher equation in physics-based diffusion-reaction systems

Mousa J. Huntul[1,*], **Serkan Alagoz**[2], **and Berat Karaagac**[3,*]

[1] Department of Mathematics, College of Science, Jazan University, P. O. Box 114, Jazan 45142, Saudi Arabia

[2] Department of Physics, Faculty of Science and Arts, Inonu University, Malatya 44000, Turkey

[3] Department of Natural and Mathematical Sciences, Faculty of Engineering, Tarsus University, Mersin 33400, Turkey

* **Correspondence:** Email: mhantool@jazanu.edu.sa; beratkaraagac@tarsus.edu.tr.

Abstract: In this study, we investigate the numerical solutions of the time-fractional Fisher equation by employing the collocation finite element method (FEM). The fractional derivative is considered in the Caputo sense, which provides a suitable framework for modeling memory-dependent diffusion-reaction processes. The temporal discretization is carried out using the $L1$ algorithm, while the spatial discretization is carried out using the unified hyperbolic polynomial B-spline basis within a collocation finite element framework. Two test problems are presented to demonstrate the efficiency of the proposed method, and the obtained numerical results are compared with available exact and reference solutions. The results confirm that the collocation finite element approach is a reliable and effective technique for solving fractional partial differential equations (FPDEs) arising in nonlinear diffusion-reaction models.

Keywords: time-fractional Fisher equation; unified hyperbolic basis; collocation method; Caputo derivative; stability
Mathematics Subject Classification: 65L60, 65M60, 42A15

1. Introduction

Fractional partial differential equations (FPDEs) have become an essential research area due to their ability to model a wide range of anomalous processes that cannot be adequately described by classical integer-order models. In contrast to standard derivatives, fractional derivatives account for memory and hereditary effects, making them well suited for describing diffusion, transport, and reaction processes in heterogeneous or complex media [1, 2]. Applications of FPDEs can be found in

physics, biology, chemistry, finance, engineering, and materials science [3]. As a result, the study of efficient analytical and numerical methods for fractional models has gained significant momentum over the past two decades. Among these FPDEs, the time-fractional Fisher equation, which is given as follows, is of particular importance:

$$ {}^{C}D_t^{\alpha} u(x,t) - \nu u_{xx}(x,t) - u(x,t)\left(1 - u^{\beta}(x,t)\right) = f(x,t), \tag{1.1} $$

where D_t^{α} symbolizes the fractional derivative in the sense of Caputo, β is an integer, and ν is a viscosity parameter. Originally introduced by R. A. Fisher in 1937 in the context of population genetics, the classical Fisher equation models the spread of an advantageous gene in a population [4]. The equation has since been applied to model diffusion–reaction systems in a variety of fields, including ecology, combustion theory, chemical kinetics, and epidemiology [5, 6]. The fractional extension of the Fisher equation incorporates a time-fractional derivative, usually in the Caputo sense, which accounts for nonlocal temporal effects such as memory and anomalous diffusion. This fractional form provides a more accurate representation of biological and physical processes in which the current state depends not only on local behavior but also on the entire history of the system [7–9].

Despite its theoretical and practical significance, finding closed-form analytical solutions of the time-fractional Fisher equation is extremely challenging. The nonlocal nature of fractional derivatives increases both the mathematical and computational complexity of the problem. As a consequence, numerical methods play a crucial role in exploring approximate solutions. Over the years, several numerical techniques have been proposed, including finite difference schemes [10] and spline-based methods [11–13]. While these methods have shown varying degrees of success, challenges such as stability, convergence, and high computational cost remain important considerations, particularly when dealing with nonlinear equations like the fractional Fisher model. The FEM is a powerful and versatile numerical tool widely used in solving partial differential equations due to its flexibility in handling irregular domains, nonuniform meshes, and complex boundary conditions [14]. Extending FEM to fractional problems has attracted growing interest, as it can efficiently capture the dynamics of anomalous diffusion-reaction systems [15]. Furthermore, the incorporation of collocation techniques within the FEM framework enhances the numerical accuracy by ensuring that the governing equation is satisfied at selected collocation points, thereby combining the strengths of both discretization and interpolation strategies [16, 17]. Among spline-based techniques, cubic B-spline collocation methods [12, 13] and non-polynomial spline approaches [11] have been successfully used for fractional Fisher-type equations. Recently, a range of advanced numerical methods has been developed for the time-fractional Fisher equation and its generalizations, including Legendre wavelet methods [18], non-polynomial spline schemes based on logarithmic derivatives for conformable fractional models [19], and accurate finite-difference algorithms [20]. Other notable contributions include fast parallel difference techniques [21], Gegenbauer spectral collocation methods [22], and Vieta–Lucas collocation approaches combined with non-standard finite-difference formulations [23]. In addition, high-order methods for generalized versions of the equation [24] and investigations concerning the long-time dynamics of solutions [25] have also been reported. Additionally, Kwak [26] and co-authors studied a normalized formulation of the time-fractional Fisher equation, which provides a cleaner asymptotic connection to the integer-order limit. In parallel with these numerical developments, several semi-analytical approaches have been proposed for fractional Fisher-type equations. In particular, the Temimi–Ansari method combined with the Caputo fractional

operator was employed to obtain approximate series solutions of the time-fractional Fisher equation without requiring assumptions on the nonlinear term [27, 28]. An integrated numerical framework for computing numerical approximations to the extended Fisher–Kolmogorov equation model was investigated in [29]. The fractional order of a two-dimensional parabolic inverse diffusion equation was also studied in [30]. These semi-analytical techniques complement numerical approaches by providing efficient series-based approximations, particularly useful for short-time or perturbative analysis. The present work focuses on the numerical treatment of the time-fractional Fisher equation through the uniform hyperbolic polynomial (UHP) B-spline collocation approach, which provides full discretization in both space and time and yields reliable approximate solutions across the entire simulation domain. On the other hand, spectral methods based on global basis functions such as Chebyshev or Legendre polynomials can achieve very high accuracy for smooth solutions, but they often lead to dense algebraic systems with computational costs of order $O(N^2)$ or higher. In contrast, the UHP B-spline basis used in this study extends the classical polynomial spline space by incorporating hyperbolic functions, providing greater flexibility in the approximation while preserving the local support property of spline methods. In addition, the resulting algebraic system is tridiagonal and can be solved efficiently in $O(N)$ operations. This gives the proposed approach a good balance between accuracy and computational efficiency. Although many numerical techniques have been proposed for the time-fractional Fisher equation, there is still a need for a collocation approach that can effectively combine higher-order smoothness, compact local support, and computational efficiency within a single unified framework. For this purpose, in the present study, we have applied a collocation FEM for solving the time-fractional Fisher equation. The method employs the collocation method based on the FEM for spatial discretization, with the aim that collocation points are used to enforce the satisfaction of the governing equation. The fractional-order derivative with respect to time is discretized via the $L1$ algorithm, and the nonlinear term, which yields $u(x,t)\,u^{\beta}(x,t)$, is linearized through the Rubin–Graves linearization technique. These processes yield a numerical scheme that will be used to obtain numerical solutions of the considered equation. Two numerical examples are presented to demonstrate the reliability of the method, and comparisons with available exact or reference solutions are provided to validate its performance. The main contribution of this study is the use of the UHP B-spline basis within a collocation framework. The UHP B-spline basis differs from standard polynomial B-spline bases in that its basis functions are constructed from hyperbolic polynomials, which are combinations of sinh, cosh, and polynomial functions, rather than purely polynomial pieces. This enriched construction extends the classical polynomial approximation space, providing greater flexibility in the representation of the solution. Combined with the $L1$ discretization of the Caputo derivative and the Rubin–Graves linearization of the nonlinear reaction term, the proposed scheme is unconditionally stable, near-second-order convergent in space, and computationally efficient. To the best of our knowledge, UHP B-splines have not previously been applied to this type of fractional diffusion–reaction model. The rest of the paper is organized as follows. Section 2 describes the unified hyperbolic basis functions and the collocation framework adopted for spatial discretization. Section 3 presents the application of the method to the time-fractional Fisher equation, including the mathematical framework of the collocation method and the derivation of the fully discrete scheme. Section 4 establishes the unconditional stability of the proposed scheme via the von Neumann criterion combined with mathematical induction. Section 5 provides the error analysis, deriving theoretical convergence estimates in both space and time.

Section 6 reports numerical experiments through two benchmark examples to assess the accuracy and efficiency of the proposed scheme. Finally, Section 7 concludes the paper with remarks on the applicability of the method, its limitations, and possible future research directions.

2. Collocation method with unified hyperbolic basis

The collocation method is a widely used numerical approach for solving differential equations by approximating the unknown solution using a finite series expansion and enforcing the governing equation at selected collocation points. To enhance accuracy and stability, we adopt the UHP functions as the trial space for spatial discretization.

2.1. Unified hyperbolic basis functions

The UHP B-spline is constructed as a piecewise function while preserving the key characteristics of standard B-splines. It is employed both in the numerical approximation of partial differential equations and in computer-aided geometric design, where it is valued for its ability to generate smooth and continuous curves and surfaces [31]. Let the computational domain be $\Omega = [a, b]$ with N partitions of equal length h such that $a = x_0 < x_1 < ... < x_{N-1} < x_N = b$, where the knot interval is $h = x_{m+1} - x_m$. The unified hyperbolic basis functions are defined as follows:

$$HB_m(x) = \frac{1}{p}\begin{cases} \sinh(x - x_{m-2}) - (x - x_{m-2}), & x_{m-2} \le x < x_{m-1}, \\ -2\sinh(x - x_{m-1}) - \sinh(x - x_m) & \\ +2(\cosh(h) + 1)(x - x_m) + 2h\cosh(h), & x_{m-1} \le x < x_m, \\ \sinh(x - x_m) + 2\sinh(x - x_{m+1}) & \\ -2(\cosh(h) + 1)(x - x_m) + 2h\cosh(h), & x_m \le x < x_{m+1}, \\ -\sinh(x - x_{m+2}) + (x - x_{m+2}), & x_{m+1} \le x < x_{m+2}, \\ 0, & \text{otherwise}, \end{cases} \tag{2.1}$$

where $p = 2h(\cosh(h) - 1)$. The UHP B-spline basis proposed by Lu et al. [32] extends the conventional B-splines through the inclusion of hyperbolic functions. This extension improves both smoothness and adaptability, making it particularly suitable for problems exhibiting hyperbolic characteristics. The set $\{HB_{-3}, HB_{-2}, \ldots, HB_{N-2}, HB_{N-1}\}$ forms a basis over $[a, b]$. Due to the shifted index convention adopted in (2.2), only the three non-zero basis functions HB_{m-3}, HB_{m-2}, and HB_{m-1} contribute at the grid point (x_m, t_n). Thus, the approximate solution $u_{app}(x_m, t)$ at the n-th time level is given by

$$u_{app}(x_m, t) = (u_{app})_m = \sum_{j=m-1}^{m+1} \delta_j(t)\, HB_{j-2}(x_m), \tag{2.2}$$

where $\delta_j(t)$ are time-dependent unknowns. The approximate solutions given in (2.2) and their derivatives are expressed as follows:

$$\begin{aligned} u_{app}(x_m,t) &\approx \eta_1\delta_{m-1}(t)+\eta_2\delta_m(t)+\eta_1\delta_{m+1}(t),\\ \left(u_{app}(x_m,t)\right)_x &\approx \xi_1\delta_{m-1}(t)+\xi_2\delta_{m+1}(t),\\ \left(u_{app}(x_m,t)\right)_{xx} &\approx \varsigma_1\delta_{m-1}(t)+\varsigma_2\delta_m(t)+\varsigma_1\delta_{m+1}(t), \end{aligned} \tag{2.3}$$

where the coefficients η_1, η_2, ξ_1, ξ_2, ς_1, and ς_2 are constants depending only on the mesh size h, defined as follows:

$$\begin{aligned} \eta_1 &= \tfrac{\sinh(h)-h}{p}, \quad \eta_2 = \tfrac{-2(\sinh(h)-h\cosh(h))}{p},\\ \xi_1 &= -\tfrac{1}{2h}, \qquad \xi_2 = \tfrac{1}{2h},\\ \varsigma_1 &= \tfrac{\sinh(h)}{p}, \qquad \varsigma_2 = \tfrac{-2\sinh(h)}{p}. \end{aligned} \tag{2.4}$$

Furthermore, the values of the approximate solution at boundary knots can be written as follows:

$$\begin{aligned} u_{app}(a,t) &\approx \eta_1\delta_{-1}(t)+\eta_2\delta_0(t)+\eta_1\delta_1(t)=u_a,\\ u_{app}(b,t) &\approx \eta_1\delta_{N-1}(t)+\eta_2\delta_N(t)+\eta_1\delta_{N+1}(t)=u_b, \end{aligned} \tag{2.5}$$

and

$$\begin{aligned} u_x(a,t) &\approx \xi_1\delta_{-1}(t)+\xi_2\delta_1(t)=u_c,\\ u_x(b,t) &\approx \xi_1\delta_{N-1}(t)+\xi_2\delta_{N+1}(t)=u_d. \end{aligned} \tag{2.6}$$

3. Application to the time-fractional Fisher equation

In this section, we develop a numerical scheme for the time-fractional Fisher equation using the collocation method with the unified hyperbolic basis. We begin by establishing the functional framework of the method, and then proceed to the full discretization.

3.1. Mathematical framework for the collocation method

The collocation FEM differs fundamentally from the standard Galerkin FEM: Rather than multiplying the governing equation by a test function and integrating over the domain, the collocation approach enforces the governing equation pointwise at a selected set of collocation points.

3.1.1. Regularity requirements and function spaces

Let $\Omega = [a,b] \subset \mathbb{R}$ and $J = (0,T]$. Because the collocation method requires pointwise evaluation of the solution and its second-order spatial derivative, the approximate solution must belong to a space of sufficiently smooth functions. We recall the standard Sobolev spaces

$$H^k(\Omega) = \left\{ v \in L^2(\Omega) \;\middle|\; \frac{d^j v}{dx^j} \in L^2(\Omega),\ j = 0,1,\ldots,k \right\}, \qquad k \geq 0,$$

equipped with the norm

$$\|v\|_{H^k} = \Big(\sum_{j=0}^{k} \|d^j v/dx^j\|_{L^2}^2\Big)^{1/2}.$$

In one spatial dimension, the Sobolev embedding theorem gives

$$H^k(\Omega) \hookrightarrow C^{k-1}[a,b], \qquad k \geq 1, \tag{3.1}$$

meaning that functions in $H^k(\Omega)$ are $(k-1)$-times continuously differentiable. Since the Fisher equation given in (1.1) involves the term u_{xx}, the collocation method requires that the approximate solution be at least in $C^2[a,b]$. Accordingly, we assume the exact solution satisfies

$$u(\cdot,t) \in H^3(\Omega) \hookrightarrow C^2[a,b], \quad \text{for almost every } t \in J, \tag{3.2}$$

which is sufficient for the pointwise collocation condition to be well-defined.

3.1.2. Pointwise collocation statement

The collocation principle replaces integration against test functions by direct pointwise enforcement of the governing equation. Let $\{x_m\}_{m=0}^N$ be the set of collocation points coinciding with the mesh nodes. The collocation method seeks an approximation $u_{app}(\cdot,t) \in V_h$ such that the residual of equation (1.1) vanishes at every collocation point:

$$\left[{}^C D_t^\alpha u_{app} - \nu\,(u_{app})_{xx} - u_{app}(1-u_{app}^\beta) - f\right]_{x=x_m} = 0, \quad m = 0,1,\ldots,N, \quad t \in J, \tag{3.3}$$

which is the strong-form of the equation. This form does not involve integration or test functions. The Eq (3.3) is well defined because $u_{app} \in V_h \subset C^2[a,b]$, so both u_{app} and $(u_{app})_{xx}$ can be evaluated pointwise at each x_m.

3.1.3. The UHP B-spline approximation space

The approximate solution is expressed as

$$u_{app}(x,t) = \sum_j \delta_j(t)\, HB_j(x),$$

where the approximation space is

$$V_h = \text{span}\{HB_{-3},\ HB_{-2},\ \ldots,\ HB_{N-1},\ HB_N\}.$$

The UHP B-spline basis functions HB_m defined in (2.1) satisfy $HB_m \in C^2[a,b]$ for all m (see [32]) so that

$$V_h \ \subset\ C^2[a,b] \ \subset\ H^2(\Omega) \ \subset\ H^1(\Omega).$$

Thus, every function in V_h and all its derivatives up to order two can be evaluated pointwise, which is the essential requirement for the collocation condition given in (3.3). By the standard B-spline approximation theory, the space V_h also satisfies the following optimal approximation property: For any $u \in H^3(\Omega)$,

$$\inf_{v_h \in V_h} \left\| u - v_h \right\|_{H^1(\Omega)} \ \le\ C\,h^2\,|u|_{H^3(\Omega)}, \tag{3.4}$$

where $C > 0$ is a constant independent of the mesh size h [14, 32]. The $O(h^2)$ spatial rate predicted by (3.4) is consistent with the near-second-order convergence rates observed in the numerical experiments reported in Section 6. Using the UHP B-spline basis and its local support property, we derive the algebraic system in the next subsection.

3.2. Derivation of the fully discrete scheme

Consider the one-dimensional time-fractional Fisher equation:

$$ {}^{C}D_{t}^{\alpha}u\left(x,t\right)-\nu u_{xx}\left(x,t\right)-u\left(x,t\right)\left(1-u^{\beta}\left(x,t\right)\right)=f\left(x,t\right), \tag{3.5} $$

subject to boundary and initial conditions

$$ \begin{aligned} u\left(a,t\right)&=u_{a}, \quad u\left(b,t\right)=u_{b},\\ u_{x}\left(a,t\right)&=u_{c}, \quad u_{x}\left(b,t\right)=u_{d}, t\geq 0,\\ u\left(x,0\right)&=g\left(x\right), \quad a\leq x\leq b. \end{aligned} \tag{3.6} $$

Here, ${}^{C}D_{t}^{\alpha}$ denotes the Caputo fractional derivative of order $0<\alpha\leq 1$, and ν is a viscosity parameter. Now, as a first step, in order to capture a time-based composition, the $L1$ algorithm [33] will be applied to the Caputo time-fractional derivative ${}^{C}D_{t}^{\alpha}u\left(x,t\right)$, and the Crank–Nicolson approach will be applied to the remaining terms one by one as follows:

$$ \begin{aligned} {}^{C}D_{t}^{\alpha}u&=\frac{\left(\Delta t\right)^{-\alpha}}{\Gamma\left(2-\alpha\right)}\sum_{k=0}^{n}b_{k}^{\alpha}\left(u_{m}^{n-k}-u_{m}^{n-1-k}\right),\\ u_{xx}=\frac{1}{2}\left(u_{xx}^{n+1}+u_{xx}^{n}\right),&=\frac{1}{2}\left(u^{n+1}+u^{n}\right), uu^{\beta}=\frac{1}{2}\left(\left(uu^{\beta}\right)^{n+1}+\left(uu^{\beta}\right)^{n}\right), \end{aligned} \tag{3.7} $$

where $\Gamma(\cdot)$ is the Euler gamma function and $b_{k}^{\alpha}=\left(k+1\right)^{1-\alpha}-k^{1-\alpha}$. When we substitute (3.7) into (3.5), we obtain

$$ \begin{aligned} &\frac{\left(\Delta t\right)^{-\alpha}}{\Gamma\left(2-\alpha\right)}\sum_{k=0}^{n}b_{k}^{\alpha}\left(u_{m}^{n-k}-u_{m}^{n-1-k}\right)-\nu\frac{1}{2}\left(u_{xx}^{n+1}+u_{xx}^{n}\right)-\frac{1}{2}\left(u^{n+1}+u^{n}\right)\\ &+\frac{1}{2}\left(\left(uu^{\beta}\right)^{n+1}+\left(uu^{\beta}\right)^{n}\right)=\frac{1}{2}\left(f\left(x,t_{n+1}\right)+f\left(x,t_{n}\right)\right). \end{aligned} \tag{3.8} $$

The equation given in (3.8) is nonlinear. To linearize the equation, a generalized expression of the Rubin–Graves linearization technique [34] is used. This technique is preferred because it linearizes the nonlinear term in a single explicit step without requiring iterative solvers, thereby preserving the computational efficiency of the scheme:

$$ \left(u^{\beta}u\right)^{n+1}\approx\left(u^{\beta}\right)^{n}u^{n+1}+\beta\left(u^{\beta-1}\right)^{n}u^{n}u^{n+1}-\beta\left(u^{\beta}\right)^{n}u^{n}. \tag{3.9} $$

The Rubin–Graves linearization [34] introduces a local truncation error of order $O(\Delta t)$ in the nonlinear term. However, this does not affect the overall convergence rate of the proposed method since the temporal error of the $L1$ approximation is $O((\Delta t)^{2-\alpha})$, where $2-\alpha>1$ for $\alpha\in(0,1)$ [35]. Therefore, the error arising from the linearization remains consistent with the accuracy of the time discretization. Moreover, in regions where the solution exhibits steep gradients or rapid changes, the numerical accuracy can still be improved by taking smaller time steps Δt and using a finer spatial mesh size h. Combining the Rubin–Graves linearization of the nonlinear term with the $L1$

approximation of the fractional derivative and the Crank–Nicolson discretization yields

$$\begin{aligned}&\sum_{k=0}^{n} b_k^{\alpha}\left(u_m^{n-k}-u_m^{n-1-k}\right)-\nu\frac{s}{2}\left(u_{xx}^{n+1}+u_{xx}^{n}\right)-\frac{s}{2}\left(u^{n+1}+u^{n}\right)\\&+\frac{s}{2}\left(\left(u^{\beta}\right)^{n}u^{n+1}+\beta\left(u^{\beta-1}\right)^{n}u^{n}u^{n+1}-\beta\left(u^{\beta}\right)^{n}u^{n}+\left(uu^{\beta}\right)^{n}\right)\\&=\frac{s}{2}\left(f\left(x,t_{n+1}\right)+f\left(x,t_{n}\right)\right),\end{aligned}\tag{3.10}$$

where $s=(\Delta t)^{\alpha}\Gamma(2-\alpha)$ and Δt is the temporal step size, defined as $\Delta t=T/M$, where T is the final time and M is the number of partitions of the time domain. After some rearrangements, we have

$$\begin{aligned}&\left(1-\frac{s}{2}\kappa_1\right)u^{n+1}-\left(\nu\frac{s}{2}\right)u_{xx}^{n+1}=\left(1+\frac{s}{2}\kappa_2\right)u^{n}+\left(\nu\frac{s}{2}\right)u_{xx}^{n}\\&+\frac{s}{2}\left(f\left(x,t_{n+1}\right)+f\left(x,t_{n}\right)\right)-\sum_{k=1}^{n-1}b_k^{\alpha}\left(u_m^{n+1-k}-u_m^{n-k}\right),\end{aligned}$$

where

$$\kappa_1=1-\left(u^{\beta}\right)^{n}-\beta\left(u^{\beta-1}\right)^{n}u^{n},\quad \kappa_2=1+\beta\left(u^{\beta}\right)^{n}u^{n}-\left(uu^{\beta}\right)^{n}.$$

To obtain a fully discretized scheme, we substitute the approximate solution given in (2.2) into (3.10). Thus, we have

$$\begin{aligned}&\delta_{m-1}^{n+1}\left(\eta_1-\frac{s}{2}\left(\kappa_1\eta_1+\nu\varsigma_1\right)\right)+\delta_m^{n+1}\left(\eta_2-\frac{s}{2}\left(\kappa_1\eta_2+\nu\varsigma_2\right)\right)+\delta_{m+1}^{n+1}\left(\eta_1-\frac{s}{2}\left(\kappa_1\eta_1+\nu\varsigma_1\right)\right)\\&=\delta_{m-1}^{n}\left(\eta_1+\frac{s}{2}\left(\kappa_2\eta_1+\nu\varsigma_1\right)\right)+\delta_m^{n}\left(\eta_2+\frac{s}{2}\left(\kappa_2\eta_2+\nu\varsigma_2\right)\right)+\delta_{m+1}^{n}\left(\eta_1+\frac{s}{2}\left(\kappa_2\eta_1+\nu\varsigma_1\right)\right)\\&+\frac{s}{2}\left(f\left(x,t_{n+1}\right)+f\left(x,t_{n}\right)\right)-\sum_{k=1}^{n-1}b_k^{\alpha}\left(\left(\eta_1\delta_{m-1}+\eta_2\delta_m+\eta_1\delta_{m+1}\right)^{n+1-k}-\left(\eta_1\delta_{m-1}+\eta_2\delta_m+\eta_1\delta_{m+1}\right)^{n-k}\right).\end{aligned}\tag{3.11}$$

The equation given in (3.11) is a system of linear algebraic equations for the varying values of m, from 0 to N. The system given below consists of $(N+1)$ equations and $(N+3)$ unknowns. To obtain a solvable system, we remove the two unknowns δ_{-1} and δ_{N+1} from the system using the discretization of the boundary condition given in (2.5) as $\delta_{-1}=(u_a-\eta_2\delta_0-\eta_1\delta_1)/\eta_1$ and $\delta_{N+1}=(u_b-\eta_1\delta_{N-1}-\eta_2\delta_N)/\eta_1$. After the elimination of the unknowns, we obtain a solvable linear algebraic equation system in the form $(N+1)\times(N+1)$. Moreover, the system can be solved sequentially in time using an initial vector. To obtain the initial vector, we use the discretization of the initial condition and boundary conditions given in (2.6) as follows:

$$\begin{aligned}&u(x,0)=\eta_1\delta_{m-1}+\eta_2\delta_m+\eta_1\delta_{m+1}=g(x),\ m=0,1,...,N,\\&\delta_{-1}=u_c/\xi_1-\xi_2\delta_1/\xi_1,\ \delta_{N+1}=u_d/\xi_2-\xi_1\delta_{N-1}/\xi_2.\end{aligned}\tag{3.12}$$

The system in (3.12) is in the form $(N+1)\times(N+5)$, and using the last two equations obtained from (2.6) yields a system in the form $(N+1)\times(N+1)$. When we solve the square matrix, we obtain the initial vector $\{\delta_0(0),\delta_1(0),..,\delta_{N-1}(0),\delta_N(0)\}$ for $t=0$. Thus, beginning with the initial vector, the unknowns δ_m^{n+1} can be obtained iteratively using the known values of δ_m^n until the desired time level is reached.

Algorithm 1 Collocation FEM via UHP B-Spline for the Time-Fractional Fisher Equation

Input: $[a, b]$, T, N, M, α, ν, β, $g(x)$, u_a, u_b, u_c, u_d, $f(x,t)$.
Output: Numerical solution U_m^n at all nodes and time levels.

1: Compute $h = (b-a)/N$, $\Delta t = T/M$, $s = (\Delta t)^\alpha \Gamma(2-\alpha)$, weights $b_k^\alpha = (k+1)^{1-\alpha} - k^{1-\alpha}$, and UHP basis coefficients $\eta_1, \eta_2, \xi_1, \xi_2, \varsigma_1, \varsigma_2$ from (2.4).
2: Solve the $(N+1) \times (N+1)$ system from (3.12) to obtain the initial coefficient vector $\boldsymbol{\delta}^0 = \{\delta_0(0), \delta_1(0), \ldots, \delta_N(0)\}$ after eliminating δ_{-1} and δ_{N+1} via (2.6).
3: **for** $n = 0, 1, \ldots, M-1$ **do**
4: Recover $(u_{app})_m^n = \eta_1\delta_{m-1}^n + \eta_2\delta_m^n + \eta_1\delta_{m+1}^n$ and compute κ_1, κ_2 via the Rubin–Graves linearization (3.9).
5: Assemble the coefficient matrix A and the right-hand side

$$\mathbf{b}^n = B\,\boldsymbol{\delta}^n + \tfrac{s}{2}(\mathbf{f}^{n+1} + \mathbf{f}^n) - \sum_{k=1}^{n-1} b_k^\alpha (\boldsymbol{\delta}^{n+1-k} - \boldsymbol{\delta}^{n-k})$$

from (3.11), where B contains the coefficients of $\boldsymbol{\delta}^n$ and the sum accumulates fractional memory from all past levels $\boldsymbol{\delta}^0, \ldots, \boldsymbol{\delta}^{n-1}$. Eliminate δ_{-1}^{n+1} and δ_{N+1}^{n+1} via (2.5).
6: Solve $A\,\boldsymbol{\delta}^{n+1} = \mathbf{b}^n$ and store $\boldsymbol{\delta}^{n+1}$ for use in subsequent time steps.
7: **end for**
8: Compute error norms, rate of convergence, and CPU.

4. Stability analysis

In this section, we discuss the stability of the linearized numerical scheme for the time-fractional Fisher equation using the collocation method designed with the help of the unified hyperbolic basis. The stability analysis is conducted under the assumptions $\alpha \in (0,1)$, $\nu > 0$, $h > 0$, and the physical bound $0 \le u \le 1$ on the solution of the time-fractional Fisher equation. In order to investigate stability, we combine the von Neumann criterion with mathematical induction. First of all, let us consider the linearized scheme as follows:

$$\left(1 - \tfrac{s\lambda}{2}\right)u^{n+1} - \tfrac{s\nu}{2}u_{xx}^{n+1} - \left(1 + \tfrac{s\lambda}{2}\right)u^n - \tfrac{s\nu}{2}u_{xx}^n - \sum_{k=1}^{n-1} b_k^\alpha \left(u^{n+1-k} - u^{n-k}\right) = 0. \tag{4.1}$$

Substituting the approximate solution given in (2.2) into (4.1), we obtain

$$\begin{aligned}
&\delta_{m-1}^{n+1}\left(\eta_1 - \frac{s}{2}(\eta_1\lambda + \varsigma_1\nu)\right) + \delta_m^{n+1}\left(\eta_2 - \frac{s}{2}(\eta_2\lambda + \varsigma_2\nu)\right) + \delta_{m+1}^{n+1}\left(\eta_1 - \frac{s}{2}(\eta_1\lambda + \varsigma_1\nu)\right) \\
&= \delta_{m-1}^{n}\left(\eta_1 + \frac{s}{2}(\eta_1\lambda + \varsigma_1\nu)\right) + \delta_m^{n}\left(\eta_2 + \frac{s}{2}(\eta_2\lambda + \varsigma_2\nu)\right) + \delta_{m+1}^{n}\left(\eta_1 + \frac{s}{2}(\eta_1\lambda + \varsigma_1\nu)\right) \\
&- \sum_{k=1}^{n-1} b_k^\alpha \left(\left(\delta_{m-1}^{n+1-k} + \delta_m^{n+1-k} + \delta_{m+1}^{n+1-k}\right) - \left(\delta_{m-1}^{n-k} + \delta_m^{n-k} + \delta_{m+1}^{n-k}\right)\right).
\end{aligned} \tag{4.2}$$

Since the proposed finite element approximation is expressed in terms of time-dependent hyperbolic basis coefficients, the stability analysis is conducted on the coefficient errors. Let σ_m^n denote the

perturbation associated with the coefficient δ_m^n. According to the von Neumann procedure, a harmonic mode can be written as follows:

$$\sigma_m^n = \epsilon^n \exp(im\phi), \tag{4.3}$$

where $i = \sqrt{-1}$, ϵ is the amplification factor, and $\phi \in [-\pi, \pi]$ is the phase angle. The numerical scheme is stable if $|\epsilon| \leq 1$, i.e., the error does not grow as $n \to \infty$. Substituting (4.3) into (4.2), we obtain the following equality for the numerical scheme:

$$\begin{aligned}
&\epsilon^{n+1} \exp(im\phi) \left\{ \left(\eta_1 - \frac{s}{2}(\eta_1\lambda + \varsigma_1\nu) \right) (\exp(-i\phi) + \exp(i\phi)) + \eta_2 - \frac{s}{2}(\eta_2\lambda + \varsigma_2\nu) \right\} \\
&= \epsilon^n \exp(im\phi) \left\{ \left(\eta_1 - \frac{s}{2}(\eta_1\lambda + \varsigma_1\nu) \right) (\exp(-i\phi) + \exp(i\phi)) + \eta_2 - \frac{s}{2}(\eta_2\lambda + \varsigma_2\nu) \right\} \\
&- \epsilon^n \exp(im\phi) \sum_{k=1}^{n-1} b_k^\alpha \left(\epsilon^{1-k} (\eta_1 (\exp(-i\phi) + \exp(i\phi)) + \eta_2) - \epsilon^{-k} (\eta_1 (\exp(-i\phi) + \exp(i\phi)) + \eta_2) \right).
\end{aligned} \tag{4.4}$$

Using Euler's formula and some rearrangements, we obtain

$$\begin{aligned}
&\epsilon^n \left\{ 2 \left(\eta_1 - \frac{s}{2}(\eta_1\lambda + \varsigma_1\nu) \right) \cos(\phi) + \eta_2 - \frac{s}{2}(\eta_2\lambda + \varsigma_2\nu) \right\} \\
&= \left\{ 2 \left(\eta_1 - \frac{s}{2}(\eta_1\lambda + \varsigma_1\nu) \right) \cos(\phi) + \eta_2 - \frac{s}{2}(\eta_2\lambda + \varsigma_2\nu) \right\} \\
&- (2\eta_1 \cos(\phi) + \eta_2) \sum_{k=1}^{n-1} b_k^\alpha \left(\epsilon^{1-k} - \epsilon^{-k} \right).
\end{aligned} \tag{4.5}$$

When we redefine the variables for simplicity and to discuss stability more clearly, we have

$$\epsilon^n = \frac{\kappa_1 - \kappa_2}{\kappa_1 + \kappa_2} - \kappa_1 \sum_{k=1}^{n-1} b_k^\alpha \left(\epsilon^{1-k} - \epsilon^{-k} \right), \tag{4.6}$$

where

$$\kappa_1 = 2\eta_1 \cos(\phi) + \eta_2, \ \kappa_2 = \frac{s}{2}(2(\eta_1\lambda + \varsigma_1\nu)\cos(\phi) + (\eta_2\lambda + \varsigma_2\nu)), \ \lambda = 1 - \omega^\beta.$$

Also, we can rewrite

$$\kappa_2 = \frac{s}{2}(\lambda(2\eta_1 \cos(\phi) + \eta_2) + \nu(2\varsigma_1 \cos(\phi) + \varsigma_2)) = \frac{s}{2}(\lambda\kappa_1 + \nu(2\varsigma_1 \cos(\phi) + \varsigma_2)).$$

Now, our aim is to prove that $|\epsilon| \leq 1$ using mathematical induction. So, for $n = 0$, we have

$$|\epsilon| = \left| \frac{\kappa_1 - \kappa_2}{\kappa_1 + \kappa_2} \right|, \tag{4.7}$$

and we will show that

$$\begin{aligned}
|\epsilon| \leq 1 &\Rightarrow |\kappa_1 - \kappa_2| \leq |\kappa_1 + \kappa_2| \\
&\Rightarrow \kappa_1^2 + \kappa_2^2 - 2\kappa_1\kappa_2 \leq \kappa_1^2 + \kappa_2^2 + 2\kappa_1\kappa_2 \\
&\Rightarrow -4\kappa_1\kappa_2 \leq 0 \\
&\Rightarrow \kappa_1\kappa_2 \geq 0.
\end{aligned} \tag{4.8}$$

Now, considering the relationship $\varsigma_2 = -2\varsigma_1$,

$$\begin{aligned}\kappa_1\kappa_2 &= \frac{s}{2}\left(\lambda\kappa_1 + \nu\left(2\varsigma_1\cos\left(\phi\right) + \varsigma_2\right)\right)\left(2\eta_1\cos\left(\phi\right) + \eta_2\right)\\ &= \frac{s}{2}\left(\lambda\kappa_1 + 2\varsigma_1\nu\left(\cos\left(\phi\right) - 1\right)\right)\left(2\eta_1\cos\left(\phi\right) + \eta_2\right).\end{aligned}$$

Since u represents a population density satisfying $0 \leq u \leq 1$, we have $\lambda = 1 - u^\beta \geq 0$, and thus $\lambda\kappa_1 \geq 0$. On the other hand, since $\varsigma_1 > 0$, $\nu > 0$, and $\cos\left(\phi\right) - 1 \leq 0$ for all $\phi \in [-\pi, \pi]$, the term $2\varsigma_1\nu\left(\cos\left(\phi\right) - 1\right) \leq 0$. Therefore, the worst case occurs at $\phi = \pi$, where $\cos\left(\phi\right) - 1 = -2$ and κ_1 attains its minimum value $\eta_2 - 2\eta_1$. At $\phi = \pi$, substituting

$$\eta_2 - 2\eta_1 = \frac{2\left[h\left(1 + \cosh(h)\right) - 2\sinh(h)\right]}{p} \;\; and \;\; \varsigma_1 = \frac{\sinh(h)}{p}$$

and cancelling $p > 0$, the condition $\kappa_1\kappa_2 \geq 0$ reduces to

$$\left(1 - u^\beta\right)\left[h\left(1 + \cosh(h)\right) - 2\sinh(h)\right] \geq 2\nu\sinh(h). \tag{4.9}$$

Since $h\left(1 + \cosh(h)\right) - 2\sinh(h) > 0$ for all $h > 0$, which follows from a simple convexity argument by noting that $g(h) := h(1 + \cosh(h)) - 2\sinh(h)$ satisfies $g(0) = 0$, $g''(h) = h\cosh(h) > 0$, and $1 - u^\beta \geq 0$, condition (4.9) is satisfied, and therefore $\kappa_1\kappa_2 \geq 0$, which completes the proof that $|\epsilon| \leq 1$ for $n = 0$. Now, we want to show that $\left|\epsilon^j\right| \leq 1$ for all j. From the induction hypothesis, assume that $\left|\epsilon^j\right| \leq 1$ for all $j \leq n - 1$. Thus, we have

$$|\epsilon^n|\left(\kappa_1 + \kappa_2\right) \leq |\kappa_1 - \kappa_2| + \kappa_1\sum_{k=1}^{n-1} b_k^\alpha\left(\left|\epsilon^{1-k}\right| + \left|\epsilon^{-k}\right|\right), \tag{4.10}$$

using the assumption for $j \leq n - 1$, $\left|\epsilon^{1-k}\right| \leq 1$, and $\left|\epsilon^{-k}\right| \leq 1$. Therefore,

$$\begin{aligned}&|\epsilon^n|\left(\kappa_1 + \kappa_2\right) \leq |\kappa_1 - \kappa_2| + 2\kappa_1\sum_{k=1}^{n-1} b_k^\alpha\\ &\Rightarrow |\epsilon^n|\left(\kappa_1 + \kappa_2\right) \leq |\kappa_1 - \kappa_2| + 2\kappa_1\left(n^{1-\alpha} - 1\right)\\ &\Rightarrow |\epsilon^n| \leq \frac{|\kappa_1 - \kappa_2|}{\kappa_1 + \kappa_2} + 2\frac{\kappa_1}{\kappa_1 + \kappa_2}\left(n^{1-\alpha} - 1\right).\end{aligned}$$

Since $\alpha \in (0, 1)$, we have $1 - \alpha \in (0, 1)$. Since $\kappa_1\kappa_2 \geq 0$ is proved in the base case, $\kappa_1 - \kappa_2 \leq \kappa_1 + \kappa_2$ holds. Moreover, at $T = n\Delta t$, $n^{1-\alpha} = (T/\Delta t)^{1-\alpha}$ is finite for any $\Delta t > 0$. Therefore, $|\epsilon^n| \leq 1$, which completes the induction and proves that the scheme is unconditionally stable in the von Neumann sense. The stability result established above is derived for the linearized scheme obtained via the Rubin–Graves approximation. For the full nonlinear problem, the physical constraint $0 \leq u \leq 1$ ensures that the linearization remains valid and the stability condition is preserved throughout the simulation. A rigorous nonlinear stability analysis via energy methods remains an open and interesting direction for future work.

5. Error analysis

In this section, we investigate the convergence behavior and accuracy of the proposed collocation method based on the UHP B-spline approximation in space together with the $L1$ discretization for the Caputo fractional derivative in time. Since the numerical scheme involves discretization in both temporal and spatial directions, the total approximation error is naturally composed of two different contributions. The first one originates from the approximation of the fractional derivative by the $L1$ formula, while the second one is due to the spline interpolation and collocation procedure used for the spatial variable.

5.1. Temporal error

We first consider the error arising from the discretization of the Caputo fractional derivative. Let $u(x,t)$ be the exact solution of the considered problem and assume that $u \in C^2(0,T;L^2(\Omega))$, which guarantees sufficient smoothness in time. The Caputo fractional derivative is approximated by the classical $L1$ formula. The $L1$ approximation provides a consistent and stable discretization for fractional derivatives of order $\alpha \in (0,1)$. For the temporal mesh $t^n = n\Delta t$, $n = 0,1,\ldots,M$, the truncation error of the $L1$ approximation satisfies the estimate [35]

$$\left\| {}^{C}D_t^{\alpha} u(\cdot,t^n) - \frac{(\Delta t)^{-\alpha}}{\Gamma(2-\alpha)} \sum_{k=0}^{n} b_k^{\alpha} \left(u^{n-k} - u^{n-1-k} \right) \right\|_{L^2(\Omega)} \leq C_1 \, (\Delta t)^{2-\alpha}, \tag{5.1}$$

where $C_1 > 0$ is independent of Δt. The estimate (5.1) indicates that the temporal accuracy of the proposed method depends directly on the fractional order α. Since $\alpha \in (0,1)$, the convergence order $2-\alpha$ lies between first and second order. In particular, when α approaches zero, the temporal accuracy becomes close to second order, whereas for $\alpha \to 1$, the convergence rate gradually reduces towards first order.

5.2. Spatial error

We next examine the spatial discretization error generated by the UHP B-spline collocation approximation. The numerical solution is investigated in the finite-dimensional spline space V_h spanned by the UHP B-spline basis functions. The UHP B-spline basis functions satisfy $HB_m \in C^2[a,b]$ [32], which ensures continuity of the approximate solution and its first two derivatives throughout the computational domain. For sufficiently smooth exact solutions, $u \in H^3(\Omega)$, the spline space V_h satisfies the approximation estimate

$$\inf_{v_h \in V_h} \|u - v_h\|_{L^2(\Omega)} \leq C_2 \, h^2 \, |u|_{H^3(\Omega)}, \tag{5.2}$$

where $C_2 > 0$ is independent of the mesh parameter h [14, 32]. Since the collocation solution u_{app}^n belongs to V_h and the collocation conditions enforce the governing equation pointwise at all nodes, the collocation error is bounded above by the best approximation error in V_h, making estimate 5.2 applicable to the proposed scheme. The estimate 5.2 demonstrates that the UHP B-spline approximation possesses second-order convergence in space. In addition, the local support property of the UHP B-spline plays an important role in the efficiency of the numerical method. Since each basis function affects only a small portion of the computational domain, the resulting algebraic system remains sparse and computationally efficient while maintaining a high level of accuracy.

5.3. Total error estimate

The overall error of the fully discrete collocation scheme is obtained by combining the temporal and spatial error contributions derived above. Let u_{app}^n denote the numerical approximation to the exact solution $u(\cdot, t^n)$ at time level t^n. Using the triangle inequality together with the unconditional stability established in Section 4 and the approximation properties of the $L1$ discretization and UHP B-spline basis functions, we obtain the following global error estimate:

$$\max_{1\le n\le M} \|u(\cdot, t^n) - U^n\|_{L^2(\Omega)} \le C\left[(\Delta t)^{2-\alpha} + h^2\right], \tag{5.3}$$

where $C > 0$ is a positive constant independent of both h and Δt. The estimate 5.3 clearly shows that the total error consists of two separate components: the temporal error term $(\Delta t)^{2-\alpha}$ and the spatial error term h^2. These theoretical convergence results are in very good agreement with the numerical findings reported in Section 4. In particular, the computed spatial convergence rates are observed to vary between approximately 1.47 and 1.93 as the mesh is refined, which is sufficiently close to the theoretical second-order rate predicted by estimate (5.3). The slight deviations from the ideal convergence order can be attributed to accumulated truncation effects, the influence of the fractional derivative, and the finite precision of the numerical computations. Overall, the obtained error estimates and numerical results confirm that the proposed UHP B-spline collocation method provides an accurate, stable, and reliable numerical tool for solving FPDEs.

6. Numerical results

In this section, we present the numerical performance of the FEM via the unified hyperbolic basis developed for the one-dimensional Fisher equation. To evaluate the accuracy and efficiency of the proposed scheme, two test examples are considered. For each example, we compute the error norms and investigate the convergence rates with respect to mesh refinement. Let the numerical solution at the spatial nodes be denoted by $U(x,t)$, and the exact solution by $u(x,t)$. The following error norms are employed:

$$\begin{aligned} L_2 &= \sum_{m=0}^{N} \sqrt{h\left(u(x_m,t) - U(x_m,t)\right)^2}, \quad (x,t) \in [a,b]\times[0,T], \\ L_\infty &= \max_{0\le m\le N} \left|u(x_m,t) - U(x_m,t)\right|. \end{aligned} \tag{6.1}$$

To quantify the rate of convergence, we use the relation

$$RoC = \frac{\log\left(\frac{E(h_1)}{E(h_2)}\right)}{\log\left(\frac{h_1}{h_2}\right)},$$

where $E(h)$ denotes the error corresponding to spatial mesh size h. All computations were performed in MATLAB on a personal computer equipped with 13th Gen Intel(R) Core(TM) i5-13500 processor (2.50 GHz) and 16 GB RAM.

6.1. Example 1

In the first test example, we consider the time-fractional Fisher equation given in (1.1) for $\beta = 1$ [10], and its related initial and boundary conditions are given as follows:

$$\begin{aligned} &u(0,t) = 0, \quad u(\pi,t) = 0, \\ &u_x(0,t) = t^\alpha, \quad u_x(\pi,t) = -t^\alpha, \quad t \geq 0, \\ &u(x,0) = 0, \quad 0 \leq x \leq \pi. \end{aligned}$$

The boundary conditions are derived from the known exact solutions. The exact solution and source function are given as [10]

$$u(x,t) = t^\alpha \sin(x), \; f(x,t) = \Gamma(\alpha+1)\sin(x) + \nu t^\alpha \sin(x) - t^\alpha \sin(x)\left(1 - t^\alpha \sin(x)\right).$$

In the first tables, namely Table 1, Table 2, and Table 3, we present the error norms L_∞ for the refined mesh sizes. Table 1 involves the variation for $\alpha = 0.1$ and $\alpha = 0.4$, while Table 2 presents the results for $\alpha = 0.6$ and $\alpha = 0.9$, with respect to varying time and space step sizes. It is clear from both tables that the L_∞ errors decrease consistently as the mesh is refined for all tested values of α, demonstrating the stability and accuracy of the proposed UHP B-spline collocation method across the full range $\alpha \in (0,1)$. A comparison across the four fractional orders reveals a clear and consistent pattern: The error norms increase monotonically as α increases from 0.1 to 0.9. Specifically, for $N = 320$ and $\Delta t = 0.001$, the L_∞ error is 1.587×10^{-5} for $\alpha = 0.1$, increasing to 4.932×10^{-5} for $\alpha = 0.4$, 8.060×10^{-5} for $\alpha = 0.6$, and further to 1.065×10^{-4} for $\alpha = 0.9$. This behavior is fully consistent with the theoretical error estimate $C\left[(\Delta t)^{2-\alpha} + h^2\right]$: as α approaches 0, the temporal convergence order $2 - \alpha$ approaches 2, yielding smaller errors, whereas as α approaches 1, the temporal order approaches 1, resulting in slightly larger errors for the same step sizes. Importantly, the method remains stable and convergent for all values of α. Table 3 presents a comparison between the collocation method and Ref. [10] for $\Delta t = 0.0001$. The numerical results in Table 3 are obtained using the UHP B-spline collocation basis. For the short-time simulation ($T = 0.01$), the present UHP method's L_∞ error exhibits negligible change under spatial refinement, producing nearly zero measured convergence rates. This behavior indicates saturation of the error at the tested spatial resolutions. This saturation occurs because at very short times, the temporal contribution $(\Delta t)^{2-\alpha}$ in the total error estimate $C[(\Delta t)^{2-\alpha} + h^2]$ becomes the dominant component, so further reduction of the spatial mesh size h produces no significant improvement until Δt is also reduced. For the longer-time simulations ($T = 1$), the UHP collocation scheme displays a consistent reduction in the error with mesh refinement and measured rates between approximately 1.47 and 1.93, indicating near-second-order convergence in the tested regime. By contrast, the reference [10] method shows steady rates ($0.3 - 0.6$ depending on α). Additionally, the table shows that the fractional order α influences the accuracy. As the fractional-order derivative increases, the error norms also increase.

Table 4 compares the present method with the recent fast parallel difference schemes (explicit-implicit (E-I) and fast alternating segment explicit-implicit (FASE-I)) of [21] for Example 1 under temporal refinement with $h = \pi/2001$. The L_∞ errors of the proposed UHP B-spline collocation method are comparable to or slightly smaller than those of both reference schemes for $\alpha = 0.4$ and 0.6 at final time $T = 1$.

Table 5 presents the L_∞ error norms together with the corresponding CPU times for Example 1 under various spatial and temporal discretization parameters. The results confirm that the proposed method achieves high accuracy at a reasonable computational cost. As the mesh improves by double N from 40 to 320, the CPU time gradually rises approximately from 0.24 seconds to 1.15 seconds, while the error consistently declines by more than one order of magnitude. This indicates that the proposed method achieves a good compromise between computational efficiency and numerical accuracy. In addition, the CPU times obtained for $\alpha = 0.4$ and $\alpha = 0.6$ remain very close to each other, suggesting that variations in the fractional order have only a minor influence on the overall computational cost of the scheme.

Table 1. The L_∞ error norms for Example 1 with $\alpha = 0.1$ and $\alpha = 0.4$ for different selections of N and Δt.

		$\alpha = 0.1$	
N / Δt	0.002	0.001	0.0005
40	3.97773254×10^{-4}	3.88036890×10^{-4}	3.83104266×10^{-4}
80	1.14222154×10^{-4}	1.04482362×10^{-4}	9.95478827×10^{-5}
160	4.33309366×10^{-5}	3.35902888×10^{-5}	2.86553465×10^{-5}
320	2.56079213×10^{-5}	1.58670597×10^{-5}	1.09320014×10^{-5}
		$\alpha = 0.4$	
40	4.76058458×10^{-4}	4.33085717×10^{-4}	4.11522334×10^{-4}
80	1.83691197×10^{-4}	1.40696591×10^{-4}	1.19122156×10^{-4}
160	1.10595047×10^{-4}	6.75949801×10^{-5}	4.60177846×10^{-5}
320	9.23207435×10^{-5}	4.93193111×10^{-5}	2.77414259×10^{-5}

Table 2. The L_∞ error norms for Example 1 with $\alpha = 0.6$ and $\alpha = 0.9$ for different selections of N and Δt.

		$\alpha = 0.6$	
N / Δt	0.002	0.001	0.0005
40	5.35555666×10^{-4}	4.61533091×10^{-4}	4.24360770×10^{-4}
80	2.45366682×10^{-4}	1.71301108×10^{-4}	1.34107070×10^{-4}
160	1.72814936×10^{-4}	9.87386200×10^{-5}	6.15391565×10^{-5}
320	1.54676723×10^{-4}	8.05977218×10^{-5}	4.33969024×10^{-5}
		$\alpha = 0.9$	
40	5.49406684×10^{-4}	4.57663029×10^{-4}	4.09315685×10^{-4}
80	2.81947627×10^{-4}	1.90138264×10^{-4}	1.41755962×10^{-4}
160	2.15079753×10^{-4}	1.23253970×10^{-4}	7.48629315×10^{-5}
320	1.98362593×10^{-4}	1.06532705×10^{-4}	5.81394836×10^{-5}

Table 3. Comparison of L_∞ error norms and rates of convergence (RoC) with Kumar et al. [10] for Example 1 at $T = 0.01$ and $T = 1$.

		$\alpha = 0.4$			
	N	[10]	RoC	Present method	RoC
T=0.01	50	2.6350×10^{-3}	—	3.1697×10^{-4}	—
	100	1.9993×10^{-3}	0.3983	3.0566×10^{-4}	0.0524
	150	1.7007×10^{-3}	0.3989	3.0357×10^{-4}	0.0170
	200	1.5162×10^{-3}	0.3992	3.0283×10^{-4}	0.0084
T=1	50	1.4922×10^{-2}	—	2.5387×10^{-4}	—
	100	1.1860×10^{-2}	0.3313	6.6723×10^{-5}	1.9278
	150	1.0263×10^{-2}	0.3567	3.2065×10^{-5}	1.8073
	200	9.2344×10^{-3}	0.3671	1.9934×10^{-5}	1.6522
		$\alpha = 0.6$			
	N	[10]	RoC	Present method	RoC
T=0.01	50	4.5432×10^{-4}	—	1.8456×10^{-4}	—
	100	2.9975×10^{-4}	0.8684	1.8299×10^{-4}	0.0174
	150	2.3503×10^{-4}	0.5999	1.8270×10^{-4}	0.0039
	200	1.9777×10^{-4}	0.6000	1.8260×10^{-4}	0.0019
T=1	50	7.0655×10^{-3}	—	2.5520×10^{-4}	—
	100	4.7119×10^{-3}	0.5845	6.9417×10^{-5}	1.8782
	150	3.7061×10^{-3}	0.5922	3.5012×10^{-5}	1.6880
	200	3.1232×10^{-3}	0.5948	2.2970×10^{-5}	1.4651

Table 4. Comparison of the L_∞ error norms with the E-I and FASE-I schemes of [21] for Example 1 with $h = \pi/2001$.

		[21]		Present method
α	N	E-I scheme	FASE-I scheme	L_∞
0.4	800	5.9110×10^{-5}	6.0420×10^{-5}	5.4152×10^{-5}
	1600	2.8962×10^{-5}	2.8096×10^{-5}	2.7205×10^{-5}
	3200	1.4224×10^{-5}	1.3651×10^{-5}	1.3697×10^{-5}
	6400	6.9880×10^{-6}	6.7237×10^{-6}	6.9316×10^{-6}
0.6	800	9.5185×10^{-5}	9.9626×10^{-5}	9.3263×10^{-5}
	1600	4.7294×10^{-5}	4.7648×10^{-5}	4.6816×10^{-5}
	3200	2.3522×10^{-5}	2.3463×10^{-5}	2.3520×10^{-5}
	6400	1.1169×10^{-5}	1.1646×10^{-5}	1.1849×10^{-5}

Table 5. The L_∞ error norms and CPU times (in seconds) for Example 1 with $\alpha = 0.4$ and $\alpha = 0.6$ for different values of N and Δt.

N	Δt	α	L_∞	CPU Time (s)
40	0.002	0.4	4.76058458×10^{-4}	0.2400
80	0.001	0.4	1.40696591×10^{-4}	0.5220
160	0.001	0.4	6.75949801×10^{-5}	0.7021
320	0.001	0.4	4.93193111×10^{-5}	1.1467
40	0.002	0.6	5.35555666×10^{-4}	0.2393
80	0.001	0.6	1.71301108×10^{-4}	0.5058
160	0.001	0.6	9.87386200×10^{-5}	0.6519
320	0.001	0.6	8.05977218×10^{-5}	1.1213

The behavior of the numerical solutions is presented in Figures 1–3. Figure 1 shows the numerical and exact solutions simultaneously for the values $N = 40$, $\Delta t = 0.002$, and $\alpha = 0.4$. The figure clearly shows how well the numerical solutions agree with the exact solutions. The smooth bell-shaped profile reflects the spatial diffusion dynamics described by the time-fractional Fisher equation. Figure 2 presents the absolute error graph. Figure 2 shows that the error vanishes at the boundaries, indicating that the boundary conditions are well adapted and that the error is symmetrically distributed within the interior region. This symmetric distribution is physically consistent with the symmetric nature of the exact solution $u(x,t) = t^\alpha \sin(x)$ over the domain $[0,\pi]$. Figure 3 presents a three-dimensional graph of the behavior of the numerical solutions for the values $\Delta t = 0.001$, $\alpha = 0.4$, and $N = 40$. The surface plot illustrates the gradual growth of the solution from zero initial conditions.

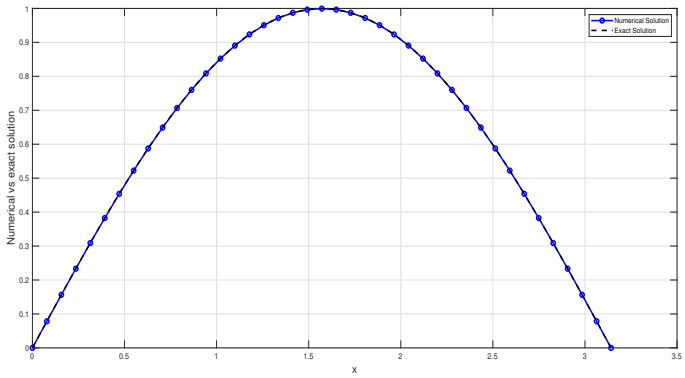

Figure 1. Numerical and exact solutions of Example 1 for $N = 40$, $\Delta t = 0.002$, and $\alpha = 0.4$ at $T = 1$.

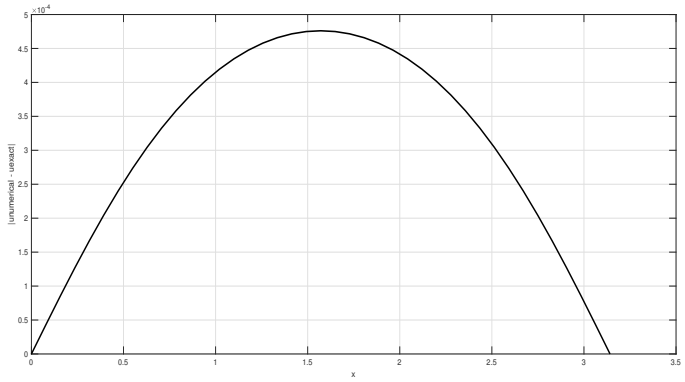

Figure 2. Absolute error graphs of Example 1 for $N = 40, \Delta t = 0.002$, and $\alpha = 0.4$ at different time levels.

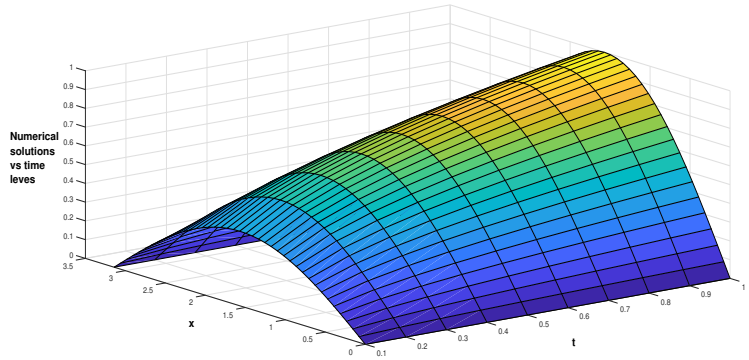

Figure 3. Three-dimensional solution behavior of Example 1 for $N = 40, \Delta t = 0.001$, and $\alpha = 0.4$.

6.2. Example 2

For the second example, we investigate the time-fractional equation for $\beta = 2$ via the following initial and boundary conditions:

$$\begin{aligned} &u(0,t) = 0, \quad u(1,t) = 0, \\ &u_x(0,t) = u_x(1,t) = 2\pi t^4, t \geq 0, \\ &u(x,0) = 0, \quad 0 \leq x \leq 1. \end{aligned}$$

The boundary conditions are obtained from the exact solution of the problem. The exact solution and source function are given as [11]

$$u(x,t) = t^4 \sin(2\pi x), \; f(x,t) = \frac{24t^{4-\alpha}}{\Gamma(5-\alpha)} \sin(2\pi x) + 4\nu \sin(2\pi x)\pi^2 t^4 - t^4 \sin(2\pi x)\left(1 - \left(t^4 \sin(2\pi x)\right)^2\right).$$

This example tests the robustness of the scheme in capturing nonlinear wave propagation. The corresponding errors are summarized in Tables 6 and 7 for various values of Δt, N, and α, covering

four fractional orders $\alpha \in \{0.1, 0.5, 0.75, 0.9\}$ that span the full range $(0, 1)$, including values close to both boundary limits. It is evident that the error decreases systematically as both the spatial resolution N and the temporal resolution Δt are refined for all tested values of α. In particular, halving Δt or doubling N leads to a notable reduction in the error, which confirms the convergence and stability of the proposed collocation finite element approach across the full range of fractional orders tested. Furthermore, the similarity of the error behavior for different values of the fractional order indicates that the method is robust with respect to the fractional order and preserves accuracy across different diffusion–reaction regimes. The results presented in Tables 6 and 7 indicate that the L_∞ errors corresponding to the tested fractional orders $\alpha \in 0.1, 0.5, 0.75, 0.9$ remain of the same magnitude and decrease steadily as the mesh is refined. These results demonstrate that the proposed method is not highly sensitive to the choice of α and preserves stable and accurate performance throughout the interval $\alpha \in (0, 1)$, including values near both endpoints.

Table 6. The L_∞ error norms for Example 2 with $\alpha = 0.1$ and $\alpha = 0.5$ for different selections of N and Δt.

		$\alpha = 0.1$	
N / Δt	0.002	0.001	0.0005
40	2.08681710×10^{-3}	2.03441649×10^{-3}	2.00786692×10^{-3}
80	6.01658614×10^{-4}	5.49125822×10^{-4}	5.22508169×10^{-4}
160	2.30009881×10^{-4}	1.77444011×10^{-4}	1.50809318×10^{-4}
320	1.37075307×10^{-4}	8.45011645×10^{-5}	5.78622109×10^{-5}
		$\alpha = 0.5$	
40	2.10444102×10^{-3}	2.02456803×10^{-3}	1.98377636×10^{-3}
80	6.48569483×10^{-4}	5.68503652×10^{-4}	5.27612576×10^{-4}
160	2.84259426×10^{-4}	2.04145343×10^{-4}	1.63229393×10^{-4}
320	1.93160578×10^{-4}	1.13034429×10^{-4}	7.21122595×10^{-5}

Table 7. The L_∞ error norms of Example 2 for different selections of N and Δt.

		$\alpha = 0.75$	
N /Δt	0.002	0.001	0.0005
40	2.08532574×10^{-3}	2.00065548×10^{-3}	1.95609296×10^{-3}
80	6.54704141×10^{-4}	5.69833987×10^{-4}	5.25165603×10^{-4}
160	2.96720137×10^{-4}	2.11799971×10^{-4}	1.67105100×10^{-4}
320	2.07203650×10^{-4}	1.22270979×10^{-4}	7.75694836×10^{-5}
		$\alpha = 0.9$	
40	2.00573155×10^{-3}	1.94977972×10^{-3}	1.91914375×10^{-3}
80	5.93229632×10^{-4}	5.37147153×10^{-4}	5.06438924×10^{-4}
160	2.39784994×10^{-4}	1.83669827×10^{-4}	1.52943519×10^{-4}
320	1.51403938×10^{-4}	9.52805967×10^{-5}	6.45497687×10^{-5}

Table 8 compares the performance of the present method with the reference scheme reported in [11] for various time levels and fractional orders. The obtained results clearly demonstrate that the collocation finite element approach yields errors several orders of magnitude smaller than those of the reference method. Table 9 provides a comparative analysis of the present method with those in [11] and [12] at different time levels for Example 2. The results show that our method consistently delivers smaller or comparable errors in both L_2 and L_∞ norms. Specifically, at $t = 0.8$, the present scheme outperforms the other approaches by several orders of magnitude, indicating its high efficiency and accuracy in long-time simulations. These findings confirm that the proposed collocation FEM is not only competitive but also advantageous over existing numerical techniques for fractional diffusion–reaction equations.

Table 8. Comparison of L_∞ and L_2 error norms with Yousif et al. [11] for Example 2 at $\alpha = 0.5$ and $\alpha = 1$.

		[11]		Present method	
α	t	L_∞	L_2	L_∞	L_2
1	0.02	9.5180×10^{-8}	9.5592×10^{-8}	4.20964963×10^{-12}	2.97667180×10^{-12}
	0.04	1.9596×10^{-7}	2.0235×10^{-7}	2.00136272×10^{-10}	1.41517715×10^{-10}
	0.06	3.0798×10^{-7}	3.2180×10^{-7}	1.42141644×10^{-9}	1.00509320×10^{-9}
	0.08	4.3676×10^{-7}	4.6040×10^{-7}	5.48982962×10^{-9}	3.88189575×10^{-9}
0.5	0.02	1.0772×10^{-7}	1.1013×10^{-7}	4.20819004×10^{-10}	2.97563971×10^{-10}
	0.04	2.1794×10^{-7}	2.2861×10^{-7}	3.04409115×10^{-9}	2.15249749×10^{-9}
	0.06	3.3310×10^{-7}	3.5237×10^{-7}	1.03733906×10^{-8}	7.33509484×10^{-9}
	0.08	4.5561×10^{-7}	4.8403×10^{-7}	2.59006231×10^{-8}	1.83145062×10^{-8}

Table 9. Comparison of L_∞ and L_2 error norms for Example 2 with Yousif et al. [11] and Majeed et al. [12] at different time levels.

t		[11]	[12]	Present method
0.6	L_∞	8.262×10^{-5}	1.97×10^{-4}	3.89288830×10^{-5}
	L_2	3.397×10^{-6}	2.541×10^{-5}	2.75278065×10^{-5}
0.8	L_∞	6.732×10^{-4}	6.366×10^{-3}	1.24633156×10^{-4}
	L_2	2.789×10^{-5}	6.371×10^{-4}	8.81590051×10^{-5}
1	L_∞	3.520×10^{-5}	3.9×10^{-4}	2.96886562×10^{-4}
	L_2	1.466×10^{-6}	5.383×10^{-5}	2.10359928×10^{-4}

Regarding computational efficiency, the CPU times reported in Tables 5 and 11 demonstrate that the proposed method completes all simulations in under 1.7 seconds for the finest mesh tested ($N = 320$), confirming its practical efficiency.

Table 11 presents the L_∞ error norms together with the corresponding CPU times for Example 2 under different mesh sizes and fractional orders. The numerical results exhibit a behavior similar to that obtained in Example 1, further demonstrating the efficiency of the proposed scheme. As the number of spatial grid points increases from $N = 40$ to $N = 320$, the CPU time grows gradually from nearly 0.13

seconds to 1.5150 seconds for $\alpha = 0.5$ and from about 0.15 seconds to 1.6361 seconds for $\alpha = 0.75$. The CPU times corresponding to $\alpha = 0.75$ are observed to be slightly larger than those obtained for $\alpha = 0.5$.

Table 10. Comparison of the proposed method with existing approaches in terms of basis type, spatial convergence order, temporal convergence order, and CPU time (in seconds).

Method	Basis	Spatial order	Temporal order	CPU (s)
Present method	UHP B-spline	≈ 2	$O((\Delta t)^{2-\alpha})$	0.13–0.64
Kumar et al. [10]	Galerkin FEM	0.33–0.60	$O((\Delta t)^{2-\alpha})$	Not reported
Yousif et al. [11]	Non-polynomial spline	$O(h^6)$	$O((\Delta t)^{2-\alpha})$	Not reported
Majeed et al. [12]	Cubic B-spline	Not reported	$O((\Delta t)^{2-\alpha})$	0.09–0.16

Table 11. The L_∞ error norms and CPU times (in seconds) for Example 2 with $\alpha = 0.5$ and $\alpha = 0.75$ for different values of N and Δt.

α	Δt	N	L_∞	CPU Time (s)
0.5	0.002	40	2.10444102×10^{-3}	0.1317
	0.001	80	5.68503652×10^{-4}	0.3599
	0.001	160	2.04145343×10^{-4}	0.7402
	0.001	320	1.13034429×10^{-4}	1.5150
0.75	0.002	40	2.08532574×10^{-3}	0.1548
	0.001	80	5.69833987×10^{-4}	0.4556
	0.001	160	2.11799971×10^{-4}	0.8266
	0.001	320	1.22270979×10^{-4}	1.6361

As in the previous example, the behavior of the numerical solutions is illustrated with figures. Figure 4 shows how closely the exact and numerical solutions agree for the values

$$N = 40; \Delta t = 0.002; \alpha = 0.5.$$

The solution of Example 2,

$$u(x, t) = t^4 \sin(2\pi x),$$

represents a spatially oscillatory profile that grows in time. As in the previous case, the proposed method achieves high-order accuracy, and the numerical solutions exhibit excellent agreement with the exact traveling wave profile. Figure 5 presents the absolute error graph. It is clear that the maximum error occurs in the middle of the interval and the minimum error occurs at $x = 0.5$, which corresponds to the zero crossing of $\sin(2\pi x)$. The three-dimensional behavior of the solutions is shown in Figure 6 for $\Delta t = 0.001$ and $\alpha = 0.5$. The surface plot illustrates the spatial structure and the temporal growth of the solution, providing a clear visualization of the overall solution behavior governed by the time-fractional Fisher equation.

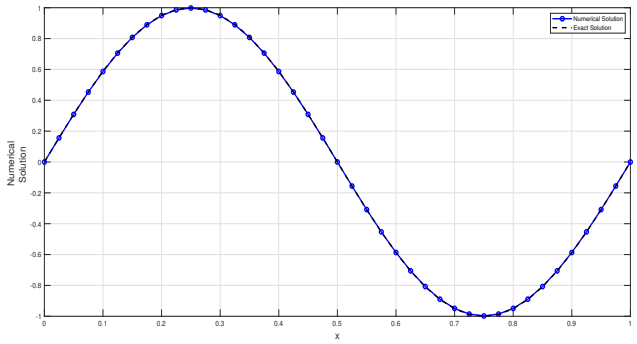

Figure 4. Numerical vs exact solutions of Example 2 for $N = 40$, $\Delta t = 0.002$, and $\alpha = 0.5$ at $T = 1$.

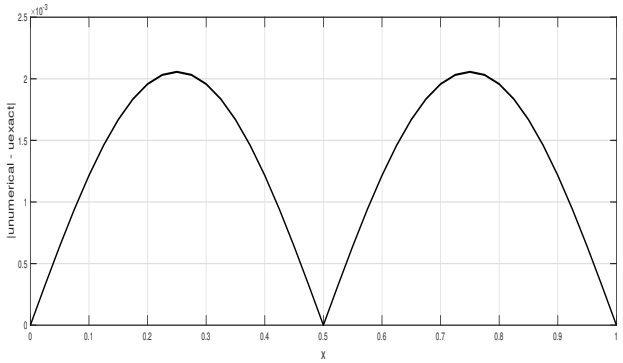

Figure 5. Graphs of absolute errors of Example 2 for $N = 40$, $\Delta t = 0.002$, and $\alpha = 0.45$ at different time levels.

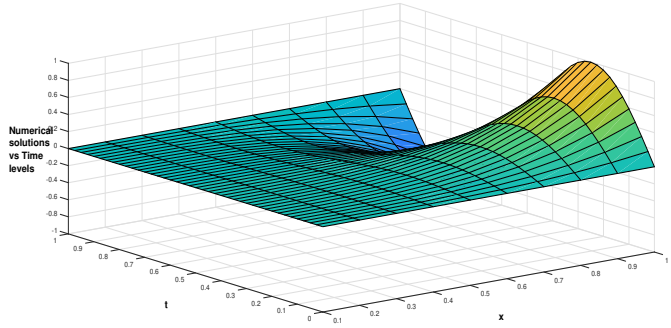

Figure 6. Three-dimensional numerical behavior of Example 2 for $N = 40$, $\Delta t = 0.001$, and $\alpha = 0.5$ at different time levels.

7. Conclusions

In this study, a collocation FEM based on unified hyperbolic basis functions was developed and analyzed for the numerical solution of the time-fractional Fisher equation with the Caputo derivative.

The temporal discretization was handled via the $L1$ algorithm, while the nonlinear reaction term was treated through the Rubin–Graves linearization technique. The stability of the proposed scheme was rigorously established using the von Neumann criterion combined with mathematical induction, confirming that the method is unconditionally stable with respect to the time step and fractional order $\alpha \in (0, 1)$. The accuracy and efficiency of the scheme were validated through two benchmark test problems, where the numerical results demonstrated excellent agreement with exact and reference solutions, and the method consistently outperformed existing approaches in terms of error norms. Furthermore, the sensitivity of the scheme to the fractional order α was investigated numerically for values across the full range $(0, 1)$, including values close to both boundary limits, confirming that the method remains stable and accurate throughout. Despite the strong performance of the proposed scheme, some limitations should be acknowledged. As with all $L1$-type discretizations of the Caputo derivative, the method requires storage of all previously computed time levels due to the nonlocal nature of the fractional operator. This memory cost can become significant for very long-time simulations.

Future research may proceed in several important directions. Although the present study is restricted to the one-dimensional case, extending the proposed method to two- and three-dimensional problems is possible. Such an extension would allow the method to be applied to a broader class of physically relevant models. In addition, the development of a rigorous nonlinear stability analysis for the fully nonlinear scheme, based on energy techniques rather than the linearized von Neumann framework employed in this study, remains an open problem. Finally, developing fast memory-efficient techniques to reduce the storage requirements associated with $L1$-type discretizations would be highly beneficial, particularly for large-scale computations and long-time simulations.

Author contributions

The authors contributed equally to this work. All authors read and approved the final manuscript.

Use of Generative-AI tools declaration

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

Acknowledgments

The authors are very grateful to the anonymous referees for their careful reading and valuable comments that led to the improvement of this manuscript.

Conflict of interest

The authors declare no conflicts of interest.

References

1. I. Podlubny, *Fractional Differential Equations*, San Diego: Academic Press, 1999. http://doi.org/10.1016/S0076-5392(99)80021-6
2. A. A. Kilbas, H. M. Srivastava, J. J. Trujillo, *Theory and Applications of Fractional Differential Equations*, Amsterdam: Elsevier, 2006. http://doi.org/10.1016/S0304-0208(06)X8001-5
3. F. Mainardi, *Fractional Calculus and Waves in Linear Viscoelasticity*, London: Imperial College Press, 2010. http://doi.org/10.1142/p614
4. R. A. Fisher, The wave of advance of advantageous genes, *Ann. Eugen.*, **7** (1937), 355–369. http://doi.org/10.1111/j.1469-1809.1937.tb02153.x
5. J. D. Murray, *Mathematical Biology I: An Introduction*, 3 Eds., New York: Springer, 2002. https://doi.org/10.1007/b98868
6. L. J. S. Allen, *An Introduction to Mathematical Biology*, Upper Saddle River: Pearson/Prentice Hall, 2006.
7. K. M. Owolabi, S. Jain, Modeling anomalous transport and pattern formation using coupled fractional reaction–diffusion equations, *Nonlinear Sci.*, **7** (2026), 100118. https://doi.org/10.1016/j.nls.2026.100118
8. K. M. Owolabi, S. Jain, E. Maré, Fractional quantum dynamics in dissipative chemical systems: Memory-driven Schrödinger models for anomalous quantum transport, *Comput. Theor. Chem.*, **1260** (2026), 115759. https://doi.org/10.1016/j.comptc.2026.115759
9. K. M. Owolabi, S. Alagoz, Advection–diffusion–reaction modeling of contaminant transport in groundwater: Analysis and simulation, *Nonlinear Sci.*, (2025), 100083. https://doi.org/10.1016/j.nls.2025.100083
10. D. Kumar, S. Chaudhary, V. V. K. S. Kumar, Fractional Crank–Nicolson–Galerkin finite element scheme for the time-fractional nonlinear diffusion equation, *Numer. Meth. PDE*, **35** (2019), 2056–2075. https://doi.org/10.1002/num.22399
11. M. A. Yousif, F. K. Hamasalh, Novel simulation of the time fractional Burgers–Fisher equations using a non-polynomial spline fractional continuity method, *AIP Adv.*, **12** (2022), 115302. http://doi.org/10.1063/5.0128819
12. A. Majeed, A. A. Zafar, S. Abbasbandy, An efficient numerical technique for solving time-fractional generalized Fisher's equation, *Front. Phys.*, **8** (2020), 293. http://doi.org/10.3389/fphy.2020.00293
13. A. Majeed, M. Kamran, M. K. Iqbal, D. Baleanu, Solving time fractional Burgers' and Fisher's equations using cubic B-spline approximation method, *Adv. Diff. Equ.*, **2020** (2020), 175. http://doi.org/10.1186/s13662-020-02619-8
14. J. N. Reddy, *An Introduction to the Finite Element Method*, 3 Eds., New York: McGraw–Hill, 2006.
15. P. Zhu, S. Xie, ADI finite element method for 2D nonlinear time fractional reaction–subdiffusion equation, *Amer. J. Comput. Math.*, **6** (2016), 336–356. http://doi.org/10.4236/ajcm.2016.64034

16. B. Karaagac, A. Esen, K. M. Owolabi, E. Pindza, A collocation method for solving time fractional nonlinear Korteweg–de Vries–Burgers equation arising in shallow water waves, *Int. J. Modern Phys. C*, **34** (2023), 2350096. https://doi.org/10.1142/S0129183123500961

17. B. Karaagac, K. M. Owolabi, A. Esen, Unveiling numerical solutions of Zeldovich model using collocation method via fourth-order uniform hyperbolic polynomial B-spline, *Adv. Theory Simul.*, **9** (2026), e01349. https://doi.org/10.1002/adts.202501349

18. F. Idiz, G. Tangolu, N. Aghazadeh, A. Mohammadi, An effective Legendre wavelet technique for the time-fractional Fisher equation, *Comput. Meth. Diff. Equ.*, **14** (2026), 145–164. http://doi.org/10.22034/cmde.2025.63725.2849

19. H. M. Srivastava, M. A. Yousif, P. O. Mohammed, T. Abdeljawad, D. Baleanu, N. Chorfi, Solving time-fractional Fisher models by non-polynomial splines in terms of logarithmic derivatives, *Fractals*, **33** (2025), 2540142. http://doi.org/https://doi.org/10.1142/S0218348X25401425

20. P. Roul, An accurate numerical method and its analysis for time-fractional Fisher's equation, *Soft Comput.*, **28** (2024), 11495–11514. https://doi.org/10.1007/s00500-024-09885-8

21. L. Chai, L. Wu, X. Yang, A fast parallel difference method for solving the time-fractional generalized Fisher equation, *J. Appl. Anal. Comput.*, **15** (2025), 1216–1240. http://doi.org/10.11948/20240159

22. A. G. Atta, Y. H. Youssri, Enhanced spectral collocation Gegenbauer approach for the time-fractional Fisher equation, *Math. Meth. Appl. Sci.*, **47** (2024), 14173–14187. https://doi.org/10.1002/mma.10263

23. M. Kashif, A Vieta–Lucas collocation and non-standard finite difference technique for solving space-time fractional-order Fisher equation, *Math. Model. Anal.*, **30** (2025), 1–16. https://doi.org/10.3846/mma.2025.19839

24. R. Choudhary, S. Singh, D. Kumar, A high-order numerical technique for generalized time-fractional Fisher's equation, *Math. Meth. Appl. Sci.*, **46** (2023), 16050–16071. https://doi.org/10.1002/mma.9435

25. A. S. Rahby, Z. Yang, Theoretical and numerical investigation of long-time behaviors for time-fractional Fisher equations, *J. Appl. Math. Comput.*, **71** (2025), 6527–6547. https://doi.org/10.1007/s12190-025-02534-9

26. S. Kwak, Y. Nam, S. Kang, J. Kim, Computational analysis of a normalized time-fractional Fisher equation, *Appl. Math. Lett.*, **166** (2025), 109542. https://doi.org/10.1016/j.aml.2025.109542

27. A. Almuneef, A. Hagag, A fractional semi-analytical iterative method for the approximate treatment of Fisher's equations, *Rev. Int. Métodos Numér. Cálc. Diseño Ing.*, **40** (2024), 50. http://doi.org/10.23967/j.rimni.2024.10.56315

28. A. El-Sayed, A. Arafa, I. Hanafy, A. Hagag, An approximate study of Fisher's equation by using a semi-analytical iterative method, *Progr. Fract. Diff. Appl.*, **9** (2023), 397–407. http://doi.org/10.18576/pfda/090305

29. M. J. Huntul, Bell polynomial-based semi-discretization approach for the extended Fisher-Kolmogorov equations, *AIMS Math.*, **10** (2025), 29263–29284. http://doi.org/10.3934/math.20251286

30. M. J. Huntul, M. Modanli, Numerical approach for solving the inverse problem: A two-dimensional time-fractional boundary value problem, *AIMS Math.*, **11** (2026), 7078–7097. http://doi.org/10.3934/math.2026291

31. M. U. Manzoor, M. Yaseen, M. Awadalla, H. Zaway, A uniform hyperbolic polynomial B-spline approach for solving the fractional diffusion-wave equations in the Caputo-Fabrizio sense, *AIMS Math.*, **10** (2025), 17049–17081 http://doi.org/10.3934/math.2025765

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

33. K. Oldham, J. Spanier, *The Fractional Calculus Theory and Applications of Differentiation and Integration to Arbitrary Order*, Amsterdam: Elsevier, 1974.

34. J. Rashidinia, M. N. Rasoulizadeh, Numerical methods based on radial basis function-generated finite difference (RBF-FD) for solution of GKdVB equation, *Wave Motion*, **90** (2019), 152–167. http://doi.org/10.1016/j.wavemoti.2019.05.006

35. C. Li, F. Zeng, *Numerical Methods for Fractional Calculus*, Boca Raton: Chapman & Hall/CRC, 2015. http://doi.org/10.1201/b18503

AIMS Press