Iranian Journal of Numerical Analysis and Optimization
Vol. 16, No. 3, 2026, pp 876–921
https://doi.org/10.22067/ijnao.2026.96568.1780
https://ijnao.um.ac.ir/

Research Article

Tension spline-quasilinearization scheme for generalized nonlinear time-fractional convection-diffusion equations

A.S. Bekela*, L.G. Enyadene and M.M. Woldaregay

Abstract

This article presents an efficient numerical technique based on a nonpolynomial tension spline for solving the generalized nonlinear time-fractional convection-diffusion equation with polynomial nonlinear convection terms. The proposed method uses nonpolynomial tension spline spatial discretization with the Crank–Nicolson time-stepping scheme, and employs the quasilinearization technique to handle the nonlinear term $p^v \frac{\partial p}{\partial x}$ for $v = 1, 2, \ldots, 10$. The tension spline introduces a controllable smoothness parameter, which can improve accuracy and reduce spurious oscillations. Additionally, the order of convergence for the suggested scheme is determined by analyzing the truncation error, and we theoretically present the stability of

*Corresponding author

Received 18 November 2025; revised 6 May 2026; accepted 9 May 2026

Alemu Senbeta Bekela
Department of Applied Mathematics, Adama Science and Technology University, Adama, Oromia, Ethiopia. e-mail: alemusenbeta1@su.edu.et

Lemi Guta Enyadene
Department of Applied Mathematics, Adama Science and Technology University, Adama, Oromia, Ethiopia. e-mail: senalemi2007@gmail.com

Mesfin Mekuria Woldaregay
Department of Applied Mathematics, Adama Science and Technology University, Adama, Oromia, Ethiopia. msfnmkr02@gmail.com

How to cite this article
Bekela, A.S. Enyadene, L.G. and Woldaregay, M.M., Tension spline-quasilinearization scheme for generalized nonlinear time-fractional convection-diffusion equations. *Iran. J. Numer. Anal. Optim.*, 2026; 16(3): 876-921. https://doi.org/10.22067/ijnao.2026.96568.1780

the developed scheme using the von Neumann method. The method is shown to be conditionally stable. The method is validated on two representative test problems, showing good agreement with expected solutions and improved accuracy compared with existing numerical methods. These results illustrate the potential of the approach for solving nonlinear time-fractional convection-diffusion problems.

AMS subject classifications (2020): 35R11, 65M06, 65M12, 65D07, 65H10.

Keywords: Caputo's fractional derivative; Crank-Nicolson scheme; Nonlinear time-fractional convection-diffusion equation; Quasi-linearization; Nonpolynomial tension spline.

1 Introduction

Fractional calculus represents a generalization of ordinary and partial differentiation and integration to arbitrary noninteger orders, thereby unifying and extending the concepts of integer-order differentiation and integration (for the history and development of this field, see, for example, [37]). Compared with classical calculus, fractional derivatives and integrals of arbitrary order provide more accurate representations of real-world phenomena [19]. Only at integer orders $(1, 2, 3, \ldots)$ can classical derivatives give information; they are unable to characterize behaviors that correspond to fractional orders $(0, 1], (1, 2], (2, 3]$ [27, 33]. On the other hand, the fractional-order derivative operator exhibits nonlocal behavior because the derivative at a given point depends on the entire history of the function, not merely its local values [5]. Unlike integer-order derivatives [13], this property improves the modeling performance of systems with memory and hereditary effects [33]. Such characteristics show that fractional derivatives are more versatile in capturing the history and hereditary properties of different dynamical systems and in modeling them [42, 1, 5].

There are numerous definitions of the fractional derivative presented in the literature (e.g., Riemann–liouville, Caputo, Weyl, Grünwald–Letnikov, and Riesz forms) [35]. Multiple definitions help guide researchers in selecting the most convenient one for obtaining the best solutions. The two most frequently utilized are the Riemann–Liouville and Caputo derivatives, which differ mainly in how integration is applied before differentiation. In order to overcome the challenge of determining initial values for the Riemann–Liouville derivative in Laplace transform applications, the Italian mathematician Caputo first proposed the Caputo fractional derivative in the 1960s [37]. The primary benefit of the Caputo approach is that it uses the same initial conditions as traditional (integer-order) differential equations. In recent studies, the Caputo derivatives have been favored over the Riemann–Liouville derivative for obtaining both exact and approximate solutions of fractional differential equations (FDEs) [35].

Solving FDEs has become essential because of their many applications and their ability to help in understanding complex real-world systems. The time-fractional convection-diffusion equation is a significant type of partial FDE, which is the focus of this investigation. It is challenging to obtain explicit analytic solutions of fractional partial differential equations (FPDEs) because of their peculiar characteristics, which include violations of the semigroup property, Leibniz rule, and chain rule [39]. As a result, FPDEs have attracted considerable interest in approximate analytical and numerical approaches. The perturbation analysis method and differential transform method [4], integral-type variational iterative method [6], and Yang transform decomposition method [7] are examples of approximate analytical methods that have been applied to solve the fractional convection-diffusion equation. One disadvantage of these techniques is that they only take into account the initial condition and overlook the boundary conditions of the fractional convection-diffusion model. Boundary conditions, however, are crucial for describing and simulating real-world processes [41]. Therefore, developing a numerical approach for solving fractional convection-diffusion models with suitable initial and boundary conditions is important.

Nonlinear FPDEs (NFPDEs) play a crucial role in understanding and modeling nonlinear behaviors that frequently occur in real-world physical and engineering systems [13, 33]. NFPDEs have recently been successfully used to apply a large number of mathematical models in mathematical biology, electrodynamics, control theory, fluid mechanics, analytical chemistry, and other fields [34, 7, 13, 25]. For studying real-time events like earthquake propagation, volcanic eruption, diffusion process, cancer models, gene models, soil dynamics in water, cosmology, and seismology, NFPDE are a major option [28, 15]. There are many nonlinear time FPDEs (NTFPDEs). For instance, the nonlinear time-fractional convection-diffusion equation (NTFCDE) [1], nonlinear time-fractional Korteweg-de Vries equation [31], nonlinear telegraph equation [16, 17], time-fractional Burgers equation, nonlinear time-fractional Klein-Gordon equation [29, 18], nonlinear time-fractional sine-Gordon equation [23], nonlinear time-fractional Schrödinger equation [43], nonlinear time-fractional reaction-diffusion equation [9, 2], and nonlinear time-fractional convection-reaction-diffusion equation [38] are the fractional versions of NTFPDEs. In this paper, we focus on the NTFCDE.

The NTFCDE, referred to as the time-fractional Burgers' equation, is one of the most important NFPDEs arising in applied sciences and mathematical physics, and it has interesting applications in physics and astrophysics [6, 13]. NTFCDEs are used in the analyze fluid mechanics modeling, diffusive wave modeling in fluid dynamics, shock wave theory, turbulent fluid mathematical modeling, sound waves in a viscous medium, and many other areas [41]. For the same reasons, recent studies have focused on examining and developing models for time-fractional Burgers (convection-diffusion) equations. Several techniques have been developed to obtain approximate or numerical solutions because many NFPDEs, including NTFCDEs, cannot be solved analytically. In this section of our manuscript, we briefly review some of them. Esen and Tasbozan [21] used the quadratic B-spline Galerkin method, which is a practical and effective finite element approach, to investigate numerical solutions of the equation. In [22], the authors obtained more accurate numerical solutions by using collocation points and increasing the degree of basis. A Shehu transform variational iteration method for solving NTFCDEs with proportional delay has been proposed by Bekela, Belachew, and Wole [6]. To obtain approximate solutions of the NTFCDE, a numerical method based on the Crank–Nicolson finite difference scheme was proposed in [35]. The formable transform Adomian decomposition method was used by Bekela and Woldaregay [10] to solve nonlinear time-fractional diffusion equations.

The NTFCDE has been discussed in the literature using a number of numerical and semi-analytic approaches. To mention a few, Li, Zhang, and Ran [32] suggested a linearized implicit finite difference scheme for the generalized NTFCDEs. The Haar wavelet scheme and cubic B-spline finite element method were introduced by Esen, Bulut, and Oruç [20]. In recent years, several numerical methods have been studied for solving the NTFCDEs, including the extended cubic B-spline method [3], the Chebyshev wavelet method [36], and combination of cubic B-splines and the differential quadrature method [26] include some of these methods. The anomalous diffusion and wave propagation models were studied in [8]. The generalized NTFCDEs or generalized time-fractional Burgers' equations, have received considerable attention in recent years and have been studied by a number of authors. Readers are referred to [13, 33, 12] and the references therein.

In this article, we consider the following generalized NTFCDE:

$$ {}_{0}^{C}D_t^{\alpha} p(x,t) + (p(x,t))^{v} \frac{\partial p(x,t)}{\partial x} = d\frac{\partial^2 p(x,t)}{\partial x^2} + f(x,t), (x,t) \in (0,L) \times (0,T), \tag{1} $$

with the initial condition

$$ p(x,0) = g(x), \quad 0 \le x \le L, \tag{2} $$

and the boundary conditions

$$ p(0,t) = g_0(t), \quad p(L,t) = g_1(t), \quad 0 \le t \le T, \tag{3} $$

where d is the diffusion coefficient parameter and v is a positive integer. Also, ${}_0^C D_t^\alpha$ is the Caputo fractional derivative of order α, where $(0 < \alpha \leq 1)$ with respect to t. The functions $g(x), g_0(t), g_1(t)$ and $f(x,t)$ are assumed to be sufficiently smooth in their corresponding domains. There are many interpretations of the concept of fractional-order derivatives, but in this paper we use the sense of Caputo; specifically,

$$ {}_0^C D_t^\alpha p(x,t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{\partial p(x,s)}{\partial s} \frac{ds}{(t-s)^\alpha}, \qquad 0 < \alpha \leq 1. \tag{4} $$

In order to solve the generalized NTFCDE, this paper proposes a conditionally stable and convergent numerical method based on a nonpolynomial tension spline. First, a sequence of linear time-fractional convection-diffusion is obtained using the quasi-linearization technique [11]. Readers are referred to [40] and [24] for additional work addressing the nonlinear term that appears in the equation using the quasi-linearization technique. Then, the time-fractional derivative is discretized using the Crank–Nicolson difference scheme. The spatial derivatives are further discretized using the nonpolynomial tension spline method, which has not been explored previously for NTFCDEs. Furthermore, some existing numerical approaches for solving NTFCDEs have a low accuracy [13, 21, 20, 35, 25, 44], and we compare the present method with results reported in those references through numerical implementation. Therefore, we have developed a hybrid numerical scheme that combines the quasi-linearization technique, the Crank–Nicolson method, and the nonpolynomial tension spline approach to achieve higher accuracy. The von Neumann analysis shows that the linearized TFPDE is conditionally stable.

The article's remaining content is arranged as follows: The linearization technique called the quasi-linearization process and its approximation are presented in Section 2. Section 3 describes the application of the nonpolynomial tension spline approach in the spatial direction and the Crank–Nicolson scheme in the temporal direction. Section 4 provides a thorough demonstration of the stability and convergence of the proposed scheme. This section also computes the local truncation error of the approximation. Numerical results are presented in Section 5 to support the theoretical findings. A brief discussion of the concluding remarks in Section 6 completes the paper.

2 Linearization technique

In this section, we linearize the generalized NTFCDE in (1) to obtain a sequence of linear time-fractional convection-diffusion equations. Then, these equations are further discretized using a nonpolynomial tension spline based on the Crank–Nicolson scheme.

2.1 Quasi-linearization technique

The method of quasi-linearization linearizes the nonlinear term in the PDE by expanding it about the solution from of the previous step in a sequence of linear TFCDEs that quadratically converge to the solution of the original nonlinear problem. Readers are referred to [11, 40] for additional information regarding the quasi-linearization process and its convergence. For the initial guess p^0, the linearization process yields a sequence $\{p^n\}$ that satisfies the problem's initial condition (2) and boundary conditions (3). We begin by defining the nonlinear operator in (1):

$$\mathcal{N}\left(p, \frac{\partial p}{\partial x}\right) = p^v \frac{\partial p}{\partial x}. \tag{5}$$

The quasi-linearization procedure uses a first-order Taylor expansion about the n^{th} iterate $\left(p^n, (\frac{\partial p}{\partial x})^n\right)$:

$$\begin{aligned}\mathcal{N}\left(p^{n+1}, \left(\frac{\partial p}{\partial x}\right)^{n+1}\right) \approx& \mathcal{N}\left(p^n, \left(\frac{\partial p}{\partial x}\right)^n\right) + \frac{\partial \mathcal{N}}{\partial p}\Big|_{\left(p^n, (\frac{\partial p}{\partial x})^n\right)} \left(p^{n+1} - p^n\right) \\ &+ \frac{\partial \mathcal{N}}{\partial p_x}\Big|_{\left(p^n, (\frac{\partial p}{\partial x})^n\right)} \left(\left(\frac{\partial p}{\partial x}\right)^{n+1} - \left(\frac{\partial p}{\partial x}\right)^n\right).\end{aligned} \tag{6}$$

After differentiation and algebraic simplification, we obtain

$$\begin{aligned}\left(p^{n+1}\right)^v \left(\frac{\partial p}{\partial x}\right)^{n+1} \approx& v\left(p^n\right)^{v-1} \left(\frac{\partial p}{\partial x}\right)^n p^{n+1} + \left(p^n\right)^v \left(\frac{\partial p}{\partial x}\right)^{n+1} \\ &- v\left(p^n\right)^v \left(\frac{\partial p}{\partial x}\right)^n.\end{aligned} \tag{7}$$

Replacing the nonlinear condition term in (1) with this linear approximation gives the iterative sequence of linear equations:

$$\begin{aligned}{}_0^C D_t^\alpha p^{n+1} =& d\left(\frac{\partial^2 p}{\partial x^2}\right)^{n+1} - v\left(p^n\right)^{v-1} \left(\frac{\partial p}{\partial x}\right)^n p^{n+1} - \left(p^n\right)^v \left(\frac{\partial p}{\partial x}\right)^{n+1} \\ &+ v\left(p^n\right)^v \left(\frac{\partial p}{\partial x}\right)^n + f^{n+1},\end{aligned} \tag{8}$$

which satisfies the original initial and boundary conditions:

$$p^{n+1}(x, 0) = g(x), \quad 0 \le x \le L, \tag{9}$$

$$p^{n+1}(0,t) = g_0(t), \quad p^{n+1}(L,t) = g_1(t), \quad 0 \le t \le T. \tag{10}$$

From (8), for simplicity of presentation, all nonlinear terms evaluated at the previous time level t_n are denoted collectively by $\bar{p}$, where $\bar{p} = p^n$. However, in the numerical analysis, these quantities are treated explicitly at their original time indices. Then, (8) becomes

$${}_0^C D_t^\alpha p^{n+1} = d\left(\frac{\partial^2 p}{\partial x^2}\right)^{n+1} - v\bar{p}^{v-1}\frac{\partial \bar{p}}{\partial x}p^{n+1} - \bar{p}^v\left(\frac{\partial p}{\partial x}\right)^{n+1} + v\bar{p}^v\frac{\partial \bar{p}}{\partial x} + f^{n+1}. \tag{11}$$

3 Numerical scheme formulation

3.1 The time semi-discretization

In this method, we first divide the spatial domain $[0, L]$ into N_x equal parts, where N_x is the number of nodal points, to obtain the partition Ω_x of the domain defined as $\Omega_x = \{x_m | x_m = mh, m = 0, 1, 2, \ldots, N_x\}$, where $h = \frac{L}{N_x}$. In the same way, the partition of the time domain $[0, T]$ is considered as $\Omega_t = \{t_n | t_n = nk, n = 0, 1, 2, \ldots, M_t\}$, where $k = \frac{T}{M_t}$. The formulas $p_m^n = p(x_m, t_n)$ and $f_m^n = f(x_m, t_n)$ represent the values of the functions p and f at the grid points (x_m, t_n), respectively. Similar to the classical Crank-Nicolson difference scheme, the Caputo fractional order α at $(x_m, t_{n+\frac{1}{2}})$ is defined as follows:

$${}_0^C D_t^\alpha p(x_m, t_{n+\frac{1}{2}}) = \frac{1}{\Gamma(1-\alpha)}\int_0^{t_{n+\frac{1}{2}}} \frac{\partial p(x_m, s)}{\partial s}\left(t_{n+\frac{1}{2}} - s\right)^{-\alpha} ds. \tag{12}$$

Let $t_n = nk$ and $t_{n+\frac{1}{2}} = \left(n + \frac{1}{2}\right)k$. Splitting the integral over the time sub-intervals yields

$$\begin{aligned}{}_0^C D_t^\alpha p(x_m, t_{n+\frac{1}{2}}) &= \frac{1}{\Gamma(1-\alpha)}\sum_{i=1}^{n}\int_{(i-1)k}^{ik} \frac{\partial p(x_m, s)}{\partial s}\left((n+\frac{1}{2})k - s\right)^{-\alpha} ds \\ &\quad + \frac{1}{\Gamma(1-\alpha)}\int_{nk}^{(n+\frac{1}{2})k} \frac{\partial p(x_m, s)}{\partial s}\left((n+\frac{1}{2})k - s\right)^{-\alpha} ds.\end{aligned}$$

For the full interval $[t_{i-1}, t_i]$, $i = 1, 2, \ldots, n$, the midpoint is $t_{i-\frac{1}{2}}$. A Taylor expansion of p_m^i and p_m^{i-1} about this midpoint is used to approximate the time derivative. Similarly, for the half-interval $[t_n, t_{n+\frac{1}{2}}]$, the midpoint is $t_{n+\frac{1}{4}}$, and a Taylor expansion of p_m^n and $p_m^{n+\frac{1}{2}}$ around this midpoint is used to approximate the derivative. By substituting these expressions, we obtain

$${}_0^C D_t^\alpha p(x_m, t_{n+\frac{1}{2}})$$

$$= \frac{1}{\Gamma(1-\alpha)} \sum_{i=1}^{n} \int_{(i-1)k}^{ik} \left(\frac{p_m^i - p_m^{i-1}}{k} + O\left(k^2\right) \right) \left(\left(n+\frac{1}{2}\right) k - s \right)^{-\alpha} ds$$
$$+ \frac{1}{\Gamma(1-\alpha)} \int_{nk}^{(n+\frac{1}{2})k} \left(\frac{p_m^{n+1} - p_m^{n}}{k} + O\left(k\right) \right) \left(\left(n+\frac{1}{2}\right) k - s \right)^{-\alpha} ds.$$

By applying the power rule of integration, we obtain

$$\begin{aligned}
&{}_0^C D_t^{\alpha} p(x_m, t_{n+\frac{1}{2}}) \\
&= \frac{1}{\Gamma(1-\alpha)} \frac{1}{(1-\alpha)k^{\alpha}} \sum_{i=1}^{n} \left(p_m^i - p_m^{i-1}\right) \left[\left(n-i+\tfrac{3}{2}\right)^{1-\alpha} - \left(n-i+\tfrac{1}{2}\right)^{1-\alpha} \right] \\
&\quad + \frac{1}{\Gamma(1-\alpha)} \frac{1}{(1-\alpha)} \sum_{i=1}^{n} \left[\left(n-i+\tfrac{3}{2}\right)^{1-\alpha} - \left(n-i+\tfrac{1}{2}\right)^{1-\alpha} \right] O\left(k^{3-\alpha}\right) \\
&\quad + \frac{1}{\Gamma(1-\alpha)} \frac{1}{(1-\alpha)k^{\alpha}} \frac{p_m^{n+1} - p_m^{n}}{2^{1-\alpha}} + \frac{1}{\Gamma(1-\alpha)} \frac{1}{(1-\alpha)} \frac{1}{2^{1-\alpha}} O\left(k^{2-\alpha}\right),
\end{aligned}$$

where $O\left(k^{3-\alpha}\right) = O\left(k^2\right) O\left(k^{1-\alpha}\right)$ and $O\left(k^{2-\alpha}\right) = O\left(k\right) O\left(k^{1-\alpha}\right)$. By rearranging, we get

$$\begin{aligned}
&{}_0^C D_t^{\alpha} p(x_m, t_{n+\frac{1}{2}}) \\
&= \frac{1}{\Gamma(2-\alpha)} \frac{1}{k^{\alpha}} \frac{p_m^{n+1} - p_m^{n}}{2^{1-\alpha}} \\
&\quad + \frac{1}{\Gamma(2-\alpha)} \frac{1}{k^{\alpha}} \sum_{i=1}^{n} \left(p_m^i - p_m^{i-1}\right) \left[\left(n-i+\tfrac{3}{2}\right)^{1-\alpha} - \left(n-i+\tfrac{1}{2}\right)^{1-\alpha} \right] \\
&\quad + \frac{1}{\Gamma(2-\alpha)} \sum_{i=1}^{n} \left[\left(n-i+\tfrac{3}{2}\right)^{1-\alpha} - \left(n-i+\tfrac{1}{2}\right)^{1-\alpha} \right] O\left(k^{3-\alpha}\right) \\
&\quad + \frac{1}{\Gamma(2-\alpha)} \frac{1}{2^{1-\alpha}} O\left(k^{2-\alpha}\right), \qquad (13)
\end{aligned}$$

where $\frac{1}{\Gamma(2-\alpha)} = \frac{1}{\Gamma(1-\alpha)} \frac{1}{(1-\alpha)}$. For computational efficiency, we re-index the sum by letting $j = n - i + 1$, so $i = n - j + 1$. With these substitutions, (13) becomes

$$\begin{aligned}
&{}_0^C D_t^{\alpha} p(x_m, t_{n+\frac{1}{2}}) \qquad (14) \\
&= \frac{1}{\Gamma(2-\alpha)} \frac{1}{k^{\alpha}} \frac{p_m^{n+1} - p_m^{n}}{2^{1-\alpha}} \\
&\quad + \frac{1}{\Gamma(2-\alpha)} \frac{1}{k^{\alpha}} \sum_{j=1}^{n} \left(p_m^{n-j+1} - p_m^{n-j}\right) \left[\left(j+\tfrac{1}{2}\right)^{1-\alpha} - \left(j-\tfrac{1}{2}\right)^{1-\alpha} \right] \\
&\quad + \frac{1}{\Gamma(2-\alpha)} \sum_{j=1}^{n} \left[\left(j+\tfrac{1}{2}\right)^{1-\alpha} - \left(j-\tfrac{1}{2}\right)^{1-\alpha} \right] O\left(k^{3-\alpha}\right) \\
&\quad + \frac{1}{\Gamma(2-\alpha)} \frac{1}{2^{1-\alpha}} O\left(k^{3-\alpha}\right). \qquad (15)
\end{aligned}$$

Let $\sigma_0 = \frac{1}{\Gamma(2-\alpha)k^\alpha}$ and $r_j = \left(j+\frac{1}{2}\right)^{1-\alpha} - (j-\frac{1}{2})^{1-\alpha}$, so that $\sum_{j=1}^n r_j = \left(n+\frac{1}{2}\right)^{1-\alpha} - \left(\frac{1}{2}\right)^{1-\alpha}$. By substituting σ_0 and r_j in (14), we get

$$\begin{aligned}
&{}_0^C D_t^\alpha p(x_m, t_{n+\frac{1}{2}}) \\
&= \sigma_0 \left[\frac{p_m^{n+1} - p_m^n}{2^{1-\alpha}} + \sum_{j=1}^n \left(p_m^{n-j+1} - p_m^{n-j}\right) r_j \right] + \mathcal{R}_1 + \mathcal{R}_2,
\end{aligned} \tag{16}$$

where

$$\begin{aligned}
&\mathcal{R}_1 + \mathcal{R}_2 \\
&= \frac{1}{\Gamma(2-\alpha)} \left[\sum_{i=1}^n r_j O\left(k^{3-\alpha}\right) + \frac{1}{2^{1-\alpha}} O\left(k^{2-\alpha}\right) \right] \\
&= \frac{1}{\Gamma(2-\alpha)} \left[\left(\left(n+\tfrac{1}{2}\right)^{1-\alpha} - \left(\tfrac{1}{2}\right)^{1-\alpha} \right) O\left(k^{3-\alpha}\right) + \left(\tfrac{1}{2}\right)^{1-\alpha} O\left(k^{2-\alpha}\right) \right].
\end{aligned}$$

Now, substituting $n = t_n/k$, we have

$$\begin{aligned}
&\mathcal{R}_1 + \mathcal{R}_2 \\
&= \frac{1}{\Gamma(2-\alpha)} \left[\left(\left(\frac{t_n}{k} + \frac{1}{2} \right)^{1-\alpha} - \left(\frac{1}{2} \right)^{1-\alpha} \right) O\left(k^{3-\alpha}\right) + \left(\tfrac{1}{2}\right)^{1-\alpha} O\left(k^{2-\alpha}\right) \right].
\end{aligned}$$

For small k, we can approximate $\left(\frac{t_n}{k} + \frac{1}{2}\right)^{1-\alpha} \approx \left(\frac{t_n}{k}\right)^{1-\alpha}$, to leading order,

$$\begin{aligned}
\mathcal{R}_1 + \mathcal{R}_2 &\approx \frac{1}{\Gamma(2-\alpha)} \left[\left(\frac{t_n}{k} \right)^{1-\alpha} O\left(k^{3-\alpha}\right) + \left(\frac{1}{2} \right)^{1-\alpha} O\left(k^{2-\alpha}\right) \right] \\
&= \frac{1}{\Gamma(2-\alpha)} \left[t_n^{1-\alpha} O\left(k^2\right) + \left(\frac{1}{2} \right)^{1-\alpha} O\left(k^{2-\alpha}\right) \right].
\end{aligned}$$

Since $0 < \alpha < 1$, we have $2 - \alpha < 2$, so $O\left(k^{2-\alpha}\right)$ dominates $O\left(k^2\right)$ for small k. Therefore,

$$\mathcal{R}_1 + \mathcal{R}_2 = O\left(k^{2-\alpha}\right). \tag{17}$$

Expressing the scheme directly in terms of the solution values p_m^i by dropping the truncation term, we rewrite the sum in (16) as

$$\begin{aligned}
&{}_0^C D_t^\alpha p(x_m, t_{n+\frac{1}{2}}) \\
&= \sigma_0 \left[\frac{p_m^{n+1} - p_m^n}{2^{1-\alpha}} + r_1 p_m^n + \sum_{i=1}^{n-1} \left(r_{n-i+1} - r_{n-i}\right) p_m^i - r_n p_m^0 \right].
\end{aligned} \tag{18}$$

Thus, (18) is the Crank–Nicolson time-stepping scheme for the Caputo time-fractional derivative at $(x_m, t_{n+\frac{1}{2}})$.

3.2 Nonpolynomial tension spline for space discretization

From space domain discretization using a uniform mesh with step size $\Delta x = h$, we have $\Omega_x = \{x_m | x_m = hm, m = 0, 1, 2, \ldots, N_x, x_{N_x} = L, h = \frac{L}{N_x}\}$, where $\Omega_x = [0, L]$. Let $\mathcal{B}(x, t, \mu)$ be the spline function associated with the tension factor μ, which belongs to the class $C^2[0, L]$. Note that when $\mu \to 0$, this parametric spline function reduced to cubic spline. To interpolate $p(x, t)$, we consider the function at the nodal point (x_m, t_n). Let $\mathcal{B}_m^n$ denote an approximation to $p_m^n = p(x_m, t_n)$, obtained from the segment $\mathcal{B}_{\Delta m}(x, t_n, \mu)$ of the interpolating function passing through the points $(x_m, \mathcal{B}_m^n)$ and $(x_{m+1}, \mathcal{B}_{m+1}^n)$. On the sub-interval $[x_m, x_{m+1}]$, the tension spline function $\mathcal{B}_\Delta(x, t_n, \mu)$ satisfies the following differential equation when $\mu > 0$:

$$\begin{aligned} &\mathcal{B}_\Delta^{''}(x, t_n, \mu) - \mu \mathcal{B}_\Delta(x, t_n, \mu) \\ &= \left[\mathcal{B}_\Delta^{''}(x_m, t_n, \mu) - \mu \mathcal{B}_\Delta(x_m, t_n, \mu)\right] \frac{x_{m+1} - x}{h} \\ &\quad + \left[\mathcal{B}_\Delta^{''}(x_{m+1}, t_n, \mu) - \mu \mathcal{B}_\Delta(x_{m+1}, t_n, \mu)\right] \frac{x - x_m}{h}, \end{aligned} \tag{19}$$

which satisfies the interpolation conditions

$$\mathcal{B}_\Delta(x_m, t_n, \mu) = p(x_m, t_n), \quad \mathcal{B}_\Delta(x_{m+1}, t_n, \mu) = p(x_{m+1}, t_n). \tag{20}$$

The spline approximation to the second-order derivative $p_{xx}(x, t)$ is given by

$$\mathcal{B}_\Delta^{''}(x_m, t_n, \mu) = M_m^n = (p_{xx})_m^n \text{ and } \mathcal{B}_\Delta^{''}(x_{m+1}, t_n, \mu) = M_{m+1}^n = (p_{xx})_{m+1}^n. \tag{21}$$

To determine the arbitrary constants, we solve (19) and apply the interpolation conditions (20), together with the spline derivative relations in (21), yielding

$$\begin{aligned} &\mathcal{B}_\Delta(x, t_n, \mu) \\ &= \frac{h^2}{\lambda^2 \sinh(\lambda)} \left[M_{m+1}^n \sinh(\lambda) \frac{x - x_m}{h} + M_m^n \sinh(\lambda) \frac{x_{m+1} - x}{h}\right] \\ &\quad - \left[\frac{h}{\lambda^2} M_m^n - \frac{1}{h^2} p_m^n\right] \frac{x_{m+1} - x}{h} - \left[\frac{h}{\lambda^2} M_{m+1}^n - \frac{1}{h^2} p_{m+1}^n\right] \frac{x - x_{m+1}}{h}, \end{aligned} \tag{22}$$

where $\lambda = \frac{h}{\sqrt{\mu}}$. Now, differentiating (22) and letting $x \to x_m$ on the interval $[x_m, x_{m+1}]$, we obtain

$$\mathcal{B}_\Delta{}'(x_m^+, t_n, \mu) = (p_x)_m^n = \frac{p_{m+1}^n - p_m^n}{h} - h\left[\gamma_1 M_{m+1}^n + \gamma_2 M_m^n\right], \tag{23}$$

where $\gamma_1 = \frac{1}{\lambda^2}\left[1 - \frac{\lambda}{\sinh(\lambda)}\right], \gamma_2 = \frac{2}{\lambda^2}\left[\lambda \coth(\lambda) - 1\right]$. By similar manner, we proceed for the interval $[x_{m-1}, x_m]$ and obtain

$$\mathcal{B}_\Delta^{'}(x_m^-, t_n, \mu) = (p_x)_m^n = \frac{p_m^n - p_{m-1}^n}{h} + h\left[\gamma_2 M_m^n + \gamma_1 M_{m-1}^n\right]. \tag{24}$$

Using the continuity of the first-order derivative of the tension spline function $\mathcal{B}_\Delta(x, t_n, \mu)$ at $x = x_m$, we obtain $\mathcal{B}_\Delta^{'}(x_m^+, t_n, \mu) = \mathcal{B}_\Delta^{'}(x_m^-, t_n, \mu)$ for $m = 1, 2, \ldots, N_x - 1, n = 0, 1, 2, \ldots, M_t$. This equality gives

$$\frac{p_{m+1}^n - p_m^n}{h} - h[\gamma_1 M_{m+1}^n + \gamma_2 M_m^n] = \frac{p_m^n - p_{m-1}^n}{h} + h[\gamma_2 M_m^n + \gamma_1 M_{m-1}^n].$$

After rearranged, this leads to the tridiagonal system

$$\gamma_1 M_{m-1}^n + \gamma_2 M_m^n + \gamma_1 M_{m+1}^n = \frac{1}{h^2}\left[p_{m-1}^n - 2p_m^n + p_{m+1}^n\right] + O\left(h^2\right). \tag{25}$$

Using the nonpolynomial spline function, we approximate the second-order spatial derivative at (x_m, t_n) [30, 14] as follows:

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_m^n = M_m^n + O(h^2). \tag{26}$$

By applying the Crank–Nicholson scheme to (11) at the point $(x_m, t_{n+\frac{1}{2}})$, we obtain

$${}_0^C D_t^\alpha p_m^{n+\frac{1}{2}} = d\left(\frac{\partial^2 p}{\partial x^2}\right)_m^{n+\frac{1}{2}} - v\bar{p}^{v-1}\frac{\partial \bar{p}}{\partial x} p_m^{n+\frac{1}{2}} - \bar{p}^v\left(\frac{\partial p}{\partial x}\right)_m^{n+\frac{1}{2}} + v\bar{p}^v\frac{\partial \bar{p}}{\partial x} + f_m^{n+\frac{1}{2}}. \tag{27}$$

Now, using (18) and (27) within the Crank–Nicolson framework, and for simplicity denoting $\left(\frac{\partial p}{\partial x}\right)_m^n = (p_x)_m^n$, $\left(\frac{\partial^2 p}{\partial x^2}\right)_m^n = (p_{xx})_m^n$, we obtain

$$\begin{aligned}
&\sigma_0\left[\frac{(p_m^{n+1} - p_m^n)}{2^{1-\alpha}} + r_1 p_m^n + \sum_{i=1}^{n}\left(r_{n-i+1} - r_{n-i}\right) p_m^i - p_m^0 r_n\right] \\
&= d\frac{(p_{xx})_m^{n+1} + (p_{xx})_m^n}{2} - v\bar{p}^{v-1}\bar{p}_x\frac{p_m^{n+1} + p_m^n}{2} - \bar{p}^v\frac{(p_x)_m^{n+1} + (p_x)_m^n}{2} \\
&\quad + v\bar{p}^v\bar{p}_x + f^{n+\frac{1}{2}}.
\end{aligned} \tag{28}$$

Using (26) in (28), we get

$$
\begin{aligned}
\frac{M_m^{n+1}+M_m^n}{2} = {} & \frac{1}{d} v \bar{p}^{v-1} \bar{p}_x \frac{p_m^{n+1}+p_m^n}{2} + \frac{1}{d} \bar{p}^v \left[\frac{(p_x)_m^{n+1}+(p_x)_m^n}{2}\right] \\
& - \frac{1}{d} v \bar{p}^v \bar{p}_x - \frac{1}{d} f_m^{n+\frac{1}{2}} \\
& + \frac{\sigma_0}{d}\left[\frac{p_m^{n+1}-p_m^n}{2^{1-\alpha}} + r_1 p_m^n + \sum_{i=1}^{n} \left(r_{n-i+1}-r_{n-i}\right) p_m^i - p_m^0 r_n\right] \\
& + \mathcal{R}_m^{n+\frac{1}{2}}.
\end{aligned} \tag{29}
$$

Rewrite (25) at $(n+1)^{th}$ time-level as

$$
\gamma_1 M_{m-1}^{n+1} + \gamma_2 M_m^{n+1} + \gamma_1 M_{m+1}^{n+1} = \frac{1}{h^2}\left[p_{m-1}^{n+1} - 2p_m^{n+1} + p_{m+1}^{n+1}\right], \tag{30}
$$

for $m = 1, 2, \ldots, N_x - 1, n = 0, 1, 2, \ldots, M_t - 1$. Adding (25) and (30), and then taking their average, we obtain

$$
\begin{aligned}
& \gamma_1 h^2\left(\frac{M_{m-1}^{n+1}+M_{m-1}^n}{2}\right) + \gamma_2 h^2\left(\frac{M_m^{n+1}+M_m^n}{2}\right) + \gamma_1 h^2\left(\frac{M_{m+1}^{n+1}+M_{m+1}^n}{2}\right) \\
& = \frac{p_{m-1}^{n+1}+p_{m-1}^n}{2} - \left(p_m^{n+1}+p_m^n\right) + \frac{p_{m+1}^{n+1}+p_{m+1}^n}{2}.
\end{aligned} \tag{31}
$$

A use of (29) in (31), yields

$$
\begin{aligned}
& \gamma_1 h^2\left[\frac{1}{d} v \bar{p}^{v-1} \bar{p}_x \frac{p_{m-1}^{n+1}+p_{m-1}^n}{2} + \frac{1}{d} \bar{p}^v \left(\frac{(p_x)_{m-1}^{n+1}+(p_x)_{m-1}^n}{2}\right)\right. \\
& - \frac{1}{d} v \bar{p}^v \bar{p}_x - \frac{1}{d} f_{m-1}^{n+\frac{1}{2}} \\
& \left. + \frac{\sigma_0}{d}\left(\frac{p_{m-1}^{n+1}-p_{m-1}^n}{2^{1-\alpha}} + r_1 p_{m-1}^n + \sum_{i=1}^{n} (r_{n-i+1}-r_{n-i}) p_{m-1}^i - p_{m-1}^0 r_n\right)\right] \\
& + \gamma_2 h^2\left[\frac{1}{d} v \bar{p}^{v-1} \bar{p}_x \frac{p_m^{n+1}+p_m^n}{2} + \frac{1}{d} \bar{p}^v \left(\frac{(p_x)_m^{n+1}+(p_x)_m^n}{2}\right) - \frac{1}{d} v \bar{p}^v \bar{p}_x - \frac{1}{d} f_m^{n+\frac{1}{2}}\right. \\
& \left. + \frac{\sigma_0}{d}\left(\frac{p_m^{n+1}-p_m^n}{2^{1-\alpha}} + r_1 p_m^n + \sum_{i=1}^{n} (r_{n-i+1}-r_{n-i}) p_m^i - p_m^0 r_n\right)\right] \\
& + \gamma_1 h^2\left[\frac{1}{d} v \bar{p}^{v-1} \bar{p}_x \frac{p_{m+1}^{n+1}+p_{m+1}^n}{2} + \frac{1}{d} \bar{p}^v \left(\frac{(p_x)_{m+1}^{n+1}+(p_x)_{m+1}^n}{2}\right)\right. \\
& - \frac{1}{d} v \bar{p}^v \bar{p}_x - \frac{1}{d} f_{m+1}^{n+\frac{1}{2}} \\
& \left. + \frac{\sigma_0}{d}\left(\frac{(p_{m+1}^{n+1}-p_{m+1}^n)}{2^{1-\alpha}} + r_1 p_{m+1}^n + \sum_{i=1}^{n} (r_{n-i+1}-r_{n-i}) p_{m+1}^i - p_{m+1}^0 r_n\right)\right]
\end{aligned}
$$

$$
= \frac{p_{m-1}^{n+1} + p_{m-1}^{n}}{2} - (p_m^{n+1} + p_m^n) + \frac{p_{m+1}^{n+1} + p_{m+1}^{n}}{2}. \tag{32}
$$

By replacing the first-order spatial derivative with the average over four neighboring points, the above equation becomes

$$
\begin{aligned}
&\gamma_1 h^2 \Bigg[\frac{1}{d} v\bar{p}^{v-1}\bar{p}_x \frac{p_{m-1}^{n+1} + p_{m-1}^{n}}{2} + \frac{1}{d}\bar{p}^v \left(\frac{p_m^{n+1} + p_m^n - p_{m-1}^{n+1} - p_{m-1}^{n}}{2h}\right) \\
&-\frac{1}{d} v\bar{p}^v \bar{p}_x - \frac{1}{d} f_{m-1}^{n+\frac{1}{2}} \\
&+\frac{\sigma_0}{d}\left(\frac{p_{m-1}^{n+1} - p_{m-1}^{n}}{2^{1-\alpha}} + r_1 p_{m-1}^{n} + \sum_{i=1}^{n}(r_{n-i+1} - r_{n-i})p_{m-1}^{i} - p_{m-1}^{0} r_n\right)\Bigg] \\
&+\gamma_2 h^2 \Bigg[\frac{1}{d} v\bar{p}^{v-1}\bar{p}_x \frac{p_m^{n+1} + p_m^n}{2} + \frac{1}{d}\bar{p}^v \left(\frac{p_{m+1}^{n+1} + p_{m+1}^{n} - p_{m-1}^{n+1} - p_{m-1}^{n}}{4h}\right) \\
&-\frac{1}{d} v\bar{p}^v \bar{p}_x - \frac{1}{d} f_{m}^{n+\frac{1}{2}} \\
&+\frac{\sigma_0}{d}\left(\frac{(p_m^{n+1} - p_m^n)}{2^{1-\alpha}} + r_1 u_m^n + \sum_{i=1}^{n}(r_{n-i+1} - r_{n-i})p_m^i - p_m^0 w_n\right)\Bigg] \\
&+\gamma_1 h^2 \Bigg[\frac{1}{d} v\bar{p}^{v-1}\bar{p}_x \frac{p_{m+1}^{n+1} + p_{m+1}^{n}}{2} + \frac{1}{d}\bar{p}^v \left(\frac{p_{m+1}^{n+1} + p_{m+1}^{n} - p_m^{n+1} - p_m^n}{2h}\right) \\
&-\frac{1}{d} v\bar{p}^v \bar{v}_x - \frac{1}{d} f_{m+1}^{n+\frac{1}{2}} \\
&+\frac{\sigma_0}{d}\left(\frac{p_{m+1}^{n+1} - p_{m+1}^{n}}{2^{1-\alpha}} + r_1 p_{m+1}^{n} + \sum_{i=1}^{n}(r_{n-i+1} - r_{n-i})p_{m+1}^{i} - p_{m+1}^{0} r_n\right)\Bigg] \\
&\quad = \frac{p_{m-1}^{n+1} + p_{m-1}^{n}}{2} - (p_m^{n+1} + p_m^n) + \frac{p_{m+1}^{n+1} + p_{m+1}^{n}}{2}.
\end{aligned} \tag{33}
$$

Equation (33) can be rearranged as

$$
\begin{aligned}
&\left[-d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} + \gamma_1 h^2 v\bar{p}^{v-1}\bar{p}_x - \gamma_1 h\bar{p}^v - \frac{\gamma_2 h}{2}\bar{p}^v\right] p_{m-1}^{n+1} \\
&+\left[2d + \frac{2\gamma_2\sigma_0 h^2}{2^{1-\alpha}} \gamma_2 h^2 v\bar{p}^{v-1}\bar{p}_x\right] p_m^{n+1} \\
&+\left[-d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} + \gamma_1 h^2 v\bar{p}^{v-1}\bar{p}_x + \gamma_1 h\bar{p}^v + \frac{\gamma_2 h}{2}\bar{p}^v\right] p_{m+1}^{n+1} \\
&\quad = \left[d + \frac{2\gamma_2\sigma_0 h^2}{2^{1-\alpha}} - \gamma_1 h^2 v\bar{p}^{v-1}\bar{p}_x + \gamma_1 h\bar{p}^v + \frac{\gamma_2 h}{2}\bar{p}^v\right] p_{m-1}^{n} \\
&\qquad + \left[-2d + \frac{2\gamma_2\sigma_0 h^2}{2^{1-\alpha}} - \gamma_2 h^2 v\bar{p}^{v-1}\bar{p}_x\right] p_m^n \\
&\qquad + \left[d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} - \gamma_1 h^2 v\bar{p}^{v-1}\bar{p}_x - \gamma_1 h\bar{p}^v - \frac{\gamma_2 h}{2}\bar{p}^v\right] p_{m+1}^{n}
\end{aligned}
$$

$$
\begin{aligned}
&-2\sigma_0 h^2 r_1(\gamma_1 p_{m-1}^n+\gamma_2 p_m^n+\gamma_1 p_{m+1}^n)\\
&+2\sigma_0 h^2 r_n(\gamma_1 p_{m-1}^0+\gamma_2 p_m^0+\gamma_1 p_{m+1}^0)\\
&-2\sigma_0 h^2\sum_{i}^{n-1}(r_{n-i+1}-r_{n-i})(\gamma_1 p_{m-1}^i+\gamma_2 p_m^i+\gamma_1 p_{m+1}^i)\\
&+4\gamma_1 h^2 v\bar{p}^v\bar{p}_x+2\gamma_2 h^2 v\bar{p}^v\bar{p}_x\\
&+h^2\left[\gamma_1 f_{m-1}^{n+\frac{1}{2}}+\gamma_2 f_m^{n+\frac{1}{2}}+\gamma_1 f_{m+1}^{n+\frac{1}{2}}\right]+2h^2\mathcal{T}_m^{n+\frac{1}{2}},
\end{aligned}
\tag{34}
$$

where $\mathcal{T}_m^{n+\frac{1}{2}}=\gamma_1\mathcal{R}_{m-1}^{n+\frac{1}{2}}+\gamma_2\mathcal{R}_m^{n+\frac{1}{2}}+\gamma_1\mathcal{R}_{m+1}^{n+\frac{1}{2}}$ denotes the truncation error $m=1,2,\dots,N_x-1$ and $n\geq 0$, which we will be proved later in section 4. Assuming that P_m^n is the approximate solution of p_m^n and removing the truncation error from (34), one obtains the following form of proposed scheme for the generalized NTFCDE in (1):

$$
\begin{aligned}
&\left[-d+\frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}}+\gamma_1 h^2 v\bar{P}^{v-1}\bar{P}_x-\gamma_1 h\bar{P}^v-\frac{\gamma_2 h}{2}\bar{P}^v\right]P_{m-1}^{n+1}\\
&+\left[2d+\frac{2\gamma_2\sigma_0 h^2}{2^{1-\alpha}}+\gamma_2 h^2 v\bar{P}^{v-1}\bar{P}_x\right]P_m^{n+1}\\
&+\left[-d+\frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}}+\gamma_1 h^2 v\bar{P}^{v-1}\bar{P}_x+\gamma_1 h\bar{P}^v+\frac{\gamma_2 h}{2}\bar{P}^v\right]P_{m+1}^{n+1}\\
&\quad=\left[d+\frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}}-\gamma_1 h^2 v\bar{P}^{v-1}\bar{P}_x+\gamma_1 h\bar{P}^v+\frac{\gamma_2 h}{2}\bar{P}^v\right]P_{m-1}^n\\
&\quad+\left[-2d+\frac{2\gamma_2\sigma_0 h^2}{2^{1-\alpha}}-\gamma_2 h^2 v\bar{P}^{v-1}\bar{P}_x\right]P_m^n\\
&\quad+\left[d+\frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}}-\gamma_1 h^2 v\bar{P}^{v-1}\bar{P}_x-\gamma_1 h\bar{P}^v-\frac{\gamma_2 h}{2}\bar{P}^v\right]P_{m+1}^n\\
&\quad-2\sigma_0 h^2 r_1(\gamma_1 P_{m-1}^n+\gamma_2 P_m^n+\gamma_1 P_{m+1}^n)\\
&\quad+2\sigma_0 h^2 r_n(\gamma_1 P_{m-1}^0+\gamma_2 P_m^0+\gamma_1 P_{m+1}^0)\\
&\quad-2\sigma_0 h^2\sum_{i}^{n-1}(r_{n-i+1}-r_{n-i})\left(\gamma_1 P_{m-1}^i+\gamma_2 P_m^i+\gamma_1 P_{m+1}^i\right)\\
&\quad+2h^2 v\bar{P}^v\bar{P}_x(2\gamma_1+\gamma_2)+h^2\left[\gamma_1 f_{m-1}^{n+\frac{1}{2}}+\gamma_2 f_m^{n+\frac{1}{2}}+\gamma_1 f_{m+1}^{n+\frac{1}{2}}\right].
\end{aligned}
\tag{35}
$$

There are N_x-1 equations with N_x+1 unknowns in the system (35). Two more equations are required in order to solve this system. From the boundary conditions in (2) and the initial condition in (3), we have

$$
\begin{aligned}
P_0^n&=g_1(t_n),P_N^n=g_2(t_n),\qquad 0\leq n\leq M_t-1,\\
P_m^0&=g(x_n),\qquad 0\leq n\leq N_x-1.
\end{aligned}
$$

With these initial and boundary conditions, (35) is written as two systems of equations for ease implementation. The system of equations for $n = 0$ is as follows:

$$AP^1 = BP^0 + C^0, \tag{36}$$

where A and B are tridiagonal matrices of order $N_x - 1$, defined as

$$A = \begin{bmatrix} b_1^1 & c_1^1 & & & & \\ a_2^1 & b_2^1 & c_2^1 & & & \\ & a_3^1 & b_3^1 & c_3^1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & a_{N_x-2}^1 & b_{N_x-2}^1 & c_{N_x-2}^1 \\ & & & & a_{N_x-1}^1 & b_{N_x-1}^1 \end{bmatrix},$$

$$B = \begin{bmatrix} b_1^0 & c_1^0 & & & & \\ a_2^0 & b_2^0 & c_2^0 & & & \\ & a_3^0 & b_3^0 & c_3^0 & & \\ & & \ddots & \ddots & \ddots & \\ & & & a_{N_x-2}^0 & b_{N_x-2}^0 & c_{Nx-2}^0 \\ & & & & a_{N_x-1}^0 & b_{N_x-1}^0 \end{bmatrix},$$

$$P^1 = \left[P_1^1, P_2^1, \cdots, P_{N_x-1}^1\right]^{\mathrm{T}}, P^0 = \left[P_1^0, P_2^0, \cdots, P_{N_x-1}^0\right]^{\mathrm{T}},$$

$$\begin{aligned} C^0 =& 2h^2 \begin{bmatrix} v\left(\bar{P}^v \bar{P}_x\right)_1^0 (2\gamma_1 + \gamma_2) + \gamma_1 f_0^{\frac{1}{2}} + \gamma_2 f_1^{\frac{1}{2}} + \gamma_1 f_2^{\frac{1}{2}} \\ v\left(\bar{P}^v \bar{P}_x\right)_2^0 (2\gamma_1 + \gamma_2) + \gamma_1 f_1^{\frac{1}{2}} + \gamma_2 f_2^{\frac{1}{2}} + \gamma_1 f_3^{\frac{1}{2}} \\ \vdots \\ v\left(\bar{P}^v \bar{P}_x\right)_{N_x-2}^0 (2\gamma_1 + \gamma_2) + \gamma_1 f_{N_x-3}^{\frac{1}{2}} + \gamma_2 f_{N_x-2}^{\frac{1}{2}} + \gamma_1 f_{N_x-1}^{\frac{1}{2}} \\ v\left(\bar{P}^v \bar{P}_x\right)_{N_x-1}^0 (2\gamma_1 + \gamma_2) + \gamma_1 f_{N_x-2}^{\frac{1}{2}} + \gamma_2 f_{N_x-1}^{\frac{1}{2}} + \gamma_1 f_{N_x}^{\frac{1}{2}} \end{bmatrix} \\ & - \begin{bmatrix} a^1 P_0^1 - a^0 P_0^0 \\ 0 \\ \vdots \\ 0 \\ c^1 P_{N_x}^1 - c^0 P_{N_x}^0 \end{bmatrix}, \end{aligned}$$

and the coefficients are

$$
\begin{aligned}
a_1 &= -d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} + \gamma_1 h^2 v \bar{P}^{v-1}\bar{P}_x - \gamma_1 h \bar{P}^v - \frac{\gamma_2 h}{2}\bar{P}^v,\\
b_1 &= 2d + \frac{2\gamma_2\sigma_0 h^2}{2^{1-\alpha}} + \gamma_2 h^2 v \bar{P}^{v-1}\bar{P}_x,\\
c_1 &= -d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} + \gamma_1 h^2 v \bar{P}^{v-1}\bar{P}_x + \gamma_1 h \bar{P}^v + \frac{\gamma_2 h}{2}\bar{P}^v,\\
a_0 &= d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} - \gamma_1 h^2 v \bar{P}^{v-1}\bar{P}_x + \gamma_1 h \bar{P}^v + \frac{\gamma_2 h}{2}\bar{P}^v,\\
b_0 &= -2d + \frac{2\gamma_2\sigma_0 h^2}{2^{1-\alpha}} - \gamma_2 h^2 v \bar{P}^{v-1}\bar{P}_x,\\
c_0 &= d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} - \gamma_1 h^2 v \bar{P}^{v-1}\bar{P}_x - \gamma_1 h \bar{P}^v - \frac{\gamma_2 h}{2}\bar{P}^v.
\end{aligned}
$$

Additionally, the following system of equations is obtained for $n \geq 1$:

$$
\begin{aligned}
AP^{n+1} = {} & BP^n + 2\sigma_0 h^2 r_n \left(\gamma_1 P^0_{m-1} + \gamma_2 P^0_m + \gamma_1 P^0_{m+1}\right) \\
& -2\sigma_0 h^2 \sum_{i}^{n-1} \left(r_{n-i+1} - r_{n-i}\right)\left(\gamma_1 P^i_{m-1} + \gamma_2 P^i_m + \gamma_1 P^i_{m+1}\right) + C^n,
\end{aligned} \tag{37}
$$

where A and B are tri-diagonal matrices of order $N_x - 1$, defined as

$$
A = \begin{bmatrix}
b_1^{n+1} & c_1^{n+1} & & & & \\
a_2^{n+1} & b_2^{n+1} & c_2^{n+1} & & & \\
 & a_3^{n+1} & b_3^{n+1} & c_3^{n+1} & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & a_{N_x-2}^{n+1} & b_{N_x-2}^{n+1} & c_{N_x-2}^{n+1} \\
 & & & & a_{N_x-1}^{n+1} & b_{N_x-1}^{n+1}
\end{bmatrix},
$$

$$
B = \begin{bmatrix}
b^n & c^n & & & & \\
a^n & b^n & c^n & & & \\
 & a^n & b^n & c^n & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & a^n & b^n & c^n \\
 & & & & a^n & b^n
\end{bmatrix},
$$

$$
C^n = 2h^2 \begin{bmatrix} v\left(\bar{P}^v \bar{P}_x\right)_1^n (2\gamma_1+\gamma_2) + \gamma_1 f_0^{n+\frac{1}{2}} + \gamma_2 f_1^{n+\frac{1}{2}} + \gamma_1 f_2^{n+\frac{1}{2}} \\ v\left(\bar{P}^v \bar{P}_x\right)_2^n (2\gamma_1+\gamma_2) + \gamma_1 f_1^{n+\frac{1}{2}} + \gamma_2 f_2^{n+\frac{1}{2}} + \gamma_1 f_3^{n+\frac{1}{2}} \\ \vdots \\ v\left(\bar{P}^v \bar{P}_x\right)_{N_x-2}^n (2\gamma_1+\gamma_2) + \gamma_1 f_{N_x-3}^{n+\frac{1}{2}} + \gamma_2 f_{N_x-2}^{n+\frac{1}{2}} + \gamma_1 f_{N_x-1}^{n+\frac{1}{2}} \\ v\left(\bar{P}^v \bar{P}_x\right)_{N_x-1}^n (2\gamma_1+\gamma_2) + \gamma_1 f_{N_x-2}^{n+\frac{1}{2}} + \gamma_2 f_{N_x-1}^{n+\frac{1}{2}} + \gamma_1 f_{N_x}^{n+\frac{1}{2}} \end{bmatrix}
- \begin{bmatrix} a^{n+1} U_0^{n+1} - a^n U_0^n \\ 0 \\ \vdots \\ 0 \\ c^{n+1} U_{N_x}^{n+1} - c^n U_{N_x}^n \end{bmatrix},
$$

and the coefficients are

$$
\begin{aligned}
a^{n+1} &= -d + \frac{2\gamma_1 \sigma_0 h^2}{2^{1-\alpha}} + \gamma_1 h^2 v \bar{P}^{v-1} \bar{P}_x - \gamma_1 h \bar{P}^v - \frac{\gamma_2 h}{2} \bar{P}^v, \\
b^{n+1} &= 2d + \frac{2\gamma_2 \sigma_0 h^2}{2^{1-\alpha}} + \gamma_2 h^2 v \bar{P}^{v-1} \bar{P}_x, \\
c^{n+1} &= -d + \frac{2\gamma_1 \sigma_0 h^2}{2^{1-\alpha}} + \gamma_1 h^2 v \bar{P}^{v-1} \bar{P}_x + \gamma_1 h \bar{P}^v + \frac{\gamma_2 h}{2} \bar{P}^v, \\
a^n &= d + \frac{2\gamma_1 \sigma_0 h^2}{2^{1-\alpha}} - \gamma_1 h^2 v \bar{P}^{v-1} \bar{P}_x + \gamma_1 h \bar{P}^v + \frac{\gamma_2 h}{2} \bar{P}^v - 2\sigma_0 h^2 r_1 \gamma_1, \\
b^n &= -2d + \frac{2\gamma_2 \sigma_0 h^2}{2^{1-\alpha}} - \gamma_2 h^2 v \bar{P}^{v-1} \bar{P}_x - 2\sigma_0 h^2 r_1 \gamma_2, \\
c^n &= d + \frac{2\gamma_1 \sigma_0 h^2}{2^{1-\alpha}} - \gamma_1 h^2 v \bar{P}^{v-1} \bar{P}_x - \gamma_1 h \bar{P}^v - \frac{\gamma_2 h}{2} \bar{P}^v - 2\sigma_0 h^2 r_1 \gamma_1.
\end{aligned}
$$

The fully discrete scheme results in a tridiagonal linear system with $N_x - 1$ unknowns at each time step. The Thomas algorithm is used to solve this system; during the forward elimination and backward substitution stages, a fixed number of arithmetic operations is required for each grid point. As a result, each time step has a computational cost of $O(N_x)$. Since the solution is advanced over M_t time steps, the algorithm's total computational substitution cost is $O(N_x M_t)$ with memory usage of order $O(N_x)$.

4 Theoretical analysis of the numerical scheme

4.1 Error analysis

Suppose that $\mathcal{T}_m$ is the local truncation error for the developed numerical scheme (25) in m^{th} term. Then,

$$\begin{aligned}\mathcal{T}_m = &\frac{1}{h^2}\left[p^n_{m-1}-2p^n_m+p^n_{m+1}\right]-\gamma_1 M^n_{m-1}-\gamma_2 M^n_m\\ &-\gamma_1 M^n_{m+1}+O\left(h^z\right), \qquad z>0.\end{aligned} \tag{38}$$

Replace M^n_{m-1}, M^n_m and M^n_{m+1} with $\frac{\partial^2 p^n_{m-1}}{\partial x^2}, \frac{\partial^2 p^n_m}{\partial x^2}$ and $\frac{\partial^2 p^n_{m+1}}{\partial x^2}$, respectively, in (38) to get

$$\begin{aligned}\mathcal{T}_m = &\frac{1}{h^2}\left[p^n_{m-1}-2p^n_m+p^n_{m+1}\right]-\gamma_1\frac{\partial^2 p^n_{m-1}}{\partial x^2}-\gamma_2\frac{\partial^2 p^n_m}{\partial x^2}\\ &-\gamma_1\frac{\partial^2 p^n_{m+1}}{\partial x^2}+O\left(h^z\right).\end{aligned} \tag{39}$$

Applying the Taylor series expansion for p^n_{m-1}, p^n_{m+1} in terms of p^n_m and its derivatives, we get

$$\begin{aligned}\mathcal{T}_m = &\left(2\gamma_1+\gamma_2-1\right)\frac{\partial^2 p^n_m}{\partial x^2}+h^2\left(\gamma_1-\frac{1}{12}\right)\frac{\partial^4 p^n_m}{\partial x^4}\\ &+h^4\left(\frac{\gamma_1}{12}-\frac{1}{360}\right)\frac{\partial^6 p^n_m}{\partial x^6}+O\left(h^z\right).\end{aligned} \tag{40}$$

It is clear from (40) that for suitable values of the parameters $\gamma_1(\gamma_1=\frac{1}{12})$ and $\gamma_2=\frac{10}{12}$ such that $2\gamma_1+\gamma_2=1$, the value of z is 2. From the local substitution truncation error $\mathcal{R}^{n+\frac{1}{2}}_m$ in (29), we have

$$\begin{aligned}\mathcal{R}^{n+1/2}_m =&M^{n+1/2}_m-\frac{1}{d}v\bar{p}^{v-1}\bar{p}_x\frac{p^{n+1}_m+p^n_m}{2}-\frac{1}{d}\bar{p}^v\left[\frac{(p_x)^{n+1}_m+(p_x)^n_m}{2}\right]\\ &+\frac{1}{d}v\bar{p}^v\bar{p}_x+\frac{1}{d}f^{n+1/2}_m\\ &-\frac{\sigma_0}{d}\left[\frac{p^{n+1}_m-p^n_m}{2^{1-\alpha}}+r_1p^n_m+\sum_{i=1}^{n}(r_{n-i+1}-r_{n-i})p^i_m-p^0_m r_n\right]\\ =&O\left(h^2\right)+O\left(k^{2-\alpha}\right)=O\left(h^2+k^{2-\alpha}\right).\end{aligned} \tag{41}$$

Based on the above concepts, truncation error $\mathcal{T}^{n+\frac{1}{2}}_m$ of (34) is given by

$$\mathcal{T}^{n+\frac{1}{2}}_m$$

$$
\begin{aligned}
&= \gamma_1 \mathcal{R}_{m-1}^{n+\frac{1}{2}} + \gamma_2 \mathcal{R}_{m}^{n+\frac{1}{2}} + \gamma_1 \mathcal{R}_{m+1}^{n+\frac{1}{2}} \\
&= -\frac{\sigma_0\gamma_1}{d}\left[\frac{p_{m-1}^{n+1}-p_{m-1}^{n}}{2^{1-\alpha}} + r_1 p_{m-1}^{n} + \sum_{i=1}^{n}(r_{n-i+1}-r_{n-i})p_{m-1}^{i} - p_{m-1}^{0} r_n\right] \\
&\quad - \frac{\sigma_0\gamma_2}{d}\left[\frac{p_{m}^{n+1}-p_{m}^{n}}{2^{1-\alpha}} + r_1 p_{m}^{n} + \sum_{i=1}^{n}(r_{n-i+1}-r_{n-i})p_{m}^{i} - p_{m}^{0} r_n\right] \\
&\quad - \frac{\sigma_0\gamma_1}{d}\left[\frac{p_{m+1}^{n+1}-p_{m+1}^{n}}{2^{1-\alpha}} + r_1 p_{m+1}^{n} + \sum_{i=1}^{n}(r_{n-i+1}-r_{n-i})p_{m+1}^{i} - p_{m+1}^{0} r_n\right] \\
&\quad - \gamma_1\left[\frac{1}{d} v \bar{p}^{v-1}\bar{p}_x \frac{p_{m-1}^{n+1}+p_{m-1}^{n}}{2} + \frac{1}{d}\bar{p}^{v}\left[\frac{(p_x)_{m-1}^{n+1}+(p_x)_{m-1}^{n}}{2}\right] - \frac{1}{d} v \bar{p}^{v}\bar{p}_x\right] \\
&\quad - \gamma_2\left[\frac{1}{d} v \bar{p}^{v-1}\bar{p}_x \frac{p_{m}^{n+1}+p_{m}^{n}}{2} + \frac{1}{d}\bar{p}^{v}\left[\frac{(p_x)_{m}^{n+1}+(p_x)_{m}^{n}}{2}\right] - \frac{1}{d} v \bar{p}^{v}\bar{p}_x\right] \\
&\quad - \gamma_1\left[\frac{1}{d} v \bar{p}^{v-1}\bar{p}_x \frac{p_{m+1}^{n+1}+p_{m+1}^{n}}{2} + \frac{1}{d}\bar{p}^{v}\left[\frac{(p_x)_{m+1}^{n+1}+(p_x)_{m+1}^{n}}{2}\right] - \frac{1}{d} v \bar{p}^{v}\bar{p}_x\right] \\
&\quad - \left(\gamma_1 M_{m-1}^{n+\frac{1}{2}} + \gamma_2 M_{m}^{n+\frac{1}{2}} + \gamma_1 M_{m+1}^{n+\frac{1}{2}}\right) \\
&\quad + \left(\gamma_1 \frac{1}{d} f_{m-1}^{n+\frac{1}{2}} + \gamma_2 \frac{1}{d} f_{m}^{n+\frac{1}{2}} + \gamma_1 \frac{1}{d} f_{m+1}^{n+\frac{1}{2}}\right) \\
&= (2\gamma_1+\gamma_2)\left(O(h^2)+O(k^{2-\alpha})\right) = O\left(h^2+k^{2-\alpha}\right). \qquad (42)
\end{aligned}
$$

Remark 1. A numerical method is said to be consistent if $\mathcal{T}_m^{n+\frac{1}{2}} \to 0$ as $h, k \to 0$

4.2 Stability analysis

In this section, we apply Von Neumann stability analysis to investigate the stability of the proposed numerical scheme. The stability analysis begins by introducing the error term $\mathcal{E}_m^n = p_m^n - P_m^n$, for $m = 0, 1, 2, \ldots, N_x, n = 0, 1, 2, \ldots, M_t$, where P_m^n denotes the numerical approximation to the exact solution of the system defined in (35). Substituting the error expression into the numerical scheme (35), we obtain the following error equation:

$$
\begin{aligned}
&\left[-d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} + \gamma_1 h^2 v \bar{P}^{v-1}\bar{P}_x - \gamma_1 h \bar{P}^{v} - \frac{\gamma_2 h}{2}\bar{P}^{v}\right]\mathcal{E}_{m-1}^{n+1} \\
&+ \left[2d + \frac{2\gamma_2\sigma_0 h^2}{2^{1-\alpha}} + \gamma_2 h^2 v \bar{P}^{v-1}\bar{P}_x\right]\mathcal{E}_{m}^{n+1} \\
&+ \left[-d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} + \gamma_1 h^2 v \bar{P}^{v-1}\bar{P}_x + \gamma_1 h \bar{P}^{v} + \frac{\gamma_2 h}{2}\bar{P}^{v}\right]\mathcal{E}_{m+1}^{n+1}
\end{aligned}
$$

$$\begin{aligned}
&= \left[d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} - \gamma_1 h^2 v \bar{P}^{v-1}\bar{P}_x + \gamma_1 h \bar{P}^v + \frac{\gamma_2 h}{2}\bar{P}^v\right]\mathcal{E}^n_{m-1} \\
&\quad + \left[-2d + \frac{2\gamma_2\sigma_0 h^2}{2^{1-\alpha}} - \gamma_2 h^2 v \bar{P}^{v-1}\bar{P}_x\right]\mathcal{E}^n_m \\
&\quad + \left[d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} - \gamma_1 h^2 v \bar{P}^{v-1}\bar{P}_x - \gamma_1 h \bar{P}^v - \frac{\gamma_2 h}{2}\bar{P}^v\right]\mathcal{E}^n_{m+1} \\
&\quad - 2\sigma_0 h^2 r_1 \left(\gamma_1\mathcal{E}^n_{m-1} + \gamma_2\mathcal{E}^n_m + \gamma_1\mathcal{E}^n_{m+1}\right) \\
&\quad + 2\sigma_0 h^2 r_n \left(\gamma_1\mathcal{E}^0_{m-1} + \gamma_2\mathcal{E}^0_m + \gamma_1\mathcal{E}^0_{m+1}\right) \\
&\quad - 2\sigma_0 h^2 \sum_{i}^{n-1}(r_{n-i+1} - r_{n-i})\left(\gamma_1\mathcal{E}^i_{m-1} + \gamma_2\mathcal{E}^i_m + \gamma_1\mathcal{E}^i_{m+1}\right),
\end{aligned} \tag{43}$$

with $\mathcal{E}^n_0 = \mathcal{E}^n_{N_x} = 0$, for $n > 0, m = 1, 2, \dots, N_x - 1$. Consider the grid function that is defined as

$$\mathcal{E}^n(x) = \begin{cases} \mathcal{E}^n_m, & x_{m-\frac{h}{2}} \le x \le x_{m+\frac{h}{2}}, \quad 1 \le m \le N_x - 1, \\ 0, & 0 \le x \le \frac{h}{2} \text{ or } x_{L-\frac{h}{2}} \le x \le L. \end{cases}$$

Then, the Fourier series expansion of $\mathcal{E}^n(x)$ is given by

$$\mathcal{E}^n(x) = \sum_{k=-\infty}^{\infty} \eta^n(k) e^{i\frac{2\pi k x}{L}},$$

where L is right boundary, $i^2 = -1$ and

$$\eta^n(k) = \frac{1}{L}\int_0^L \mathcal{E}^n(\beta) e^{-i\frac{2\pi k\beta}{L}} d\beta, \qquad n = 1, 2, M_t,$$

are known as discrete Fourier coefficients. Defining the norm

$$\|\mathcal{E}^n\|_2 = \left[\sum_{m=1}^{N-1} h|\mathcal{E}^n_m|^2\right]^{\frac{1}{2}} = \left[\int_0^L |\mathcal{E}^n(x)|^2 dx\right]^{\frac{1}{2}}.$$

Introducing Parseval's identity (for the discrete Fourier transform)

$$\int_0^L |\mathcal{E}^n(x)|^2 dx = \sum_{k=-\infty}^{\infty} |\eta^n(k)|^2,$$

and the norm

$$\|\mathcal{E}^n\|_2 = \left(\sum_{m}^{N_x-1} h|\mathcal{E}^n_m|^2\right)^{\frac{1}{2}} = \left(\int_0^L |\mathcal{E}^n(x)^2 dx|\right)^{\frac{1}{2}}.$$

Thus, we obtain the following expression:

$$\|\mathcal{E}^n\|_2^2 = \sum_{k=-\infty}^{\infty} |\eta^n(k)|^2.$$

Based on the above analysis discussed in this section, we can write the solution of (43) as $\mathcal{E}_m^n = \eta^n e^{imh\theta}$, where $L = 1$ and $\theta = \frac{2\pi k}{L}$. Now, we use the expression $\mathcal{E}_m^n = \eta^n e^{imh\theta}$ in (43) to obtain

$$\begin{aligned}
&\Big[-d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} + \gamma_1 h^2 v\bar{P}^{v-1}\bar{P}_x - \gamma_1 h\bar{P}^v - \frac{\gamma_2 h}{2}\bar{P}^v\Big]\eta^{n+1}e^{i(m-1)h\theta}\\
&+\Big[2d + \frac{2\gamma_2\sigma_0 h^2}{2^{1-\alpha}} + \gamma_2 h^2 v\bar{P}^{v-1}\bar{P}_x\Big]\eta^{n+1}e^{imh\theta}\\
&+\Big[-d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} + \gamma_1 h^2 v\bar{P}^{v-1}\bar{P}_x + \gamma_1 h\bar{P}^v + \frac{\gamma_2 h}{2}\bar{P}^v\Big]\eta^{n+1}e^{i(m+1)h\theta}\\
&\quad = \Big[d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} - \gamma_1 h^2 v\bar{P}^{v-1}\bar{P}_x + \gamma_1 h\bar{P}^v + \frac{\gamma_2 h}{2}\bar{P}^v\Big]\eta^{n}e^{i(m-1)h\theta}\\
&\quad + \Big[-2d + \frac{2\gamma_2\sigma_0 h^2}{2^{1-\alpha}} - \gamma_2 h^2 v\bar{P}^{v-1}\bar{P}_x\Big]\eta^{n}e^{imh\theta}\\
&\quad + \Big[d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} - \gamma_1 h^2 v\bar{P}^{v-1}\bar{P}_x - \gamma_1 h\bar{P}^v - \frac{\gamma_2 h}{2}\bar{P}^v\Big]\eta^{n}e^{i(m+1)h\theta}\\
&\quad - 2\sigma h^2 r_1\left(\gamma_1 e^{i(m-1)h\theta} + \gamma_2 e^{imh\theta} + \gamma_1 e^{i(m+1)h\theta}\right)\eta^n\\
&\quad + 2\sigma h^2 r_n\left(\gamma_1 e^{i(m-1)h\theta} + \gamma_2 e^{imh\theta} + \gamma_1 e^{i(m+1)h\theta}\right)\eta^0\\
&\quad - 2\sigma_0 h^2\sum_{j=1}^{n-1}(r_{n-j+1} - r_{n-j})\left(\gamma_1 e^{i(m-1)h\theta} + \gamma_2 e^{imh\theta} + \gamma_1 e^{i(m+1)h\theta}\right)\eta^j.
\end{aligned} \tag{44}$$

Divide the entire equation by $e^{imh\theta}$ and apply the trigonometric identities $e^{ih\theta} + e^{-ih\theta} = 2\cos(h\theta), e^{ih\theta} - e^{-ih\theta} = 2i\sin(h\theta)$, we get

$$\begin{aligned}
&\Bigg[2\left(-d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} + \gamma_1 h^2 v\bar{P}^{v-1}\bar{P}_x\right)\cos(h\theta)\\
&+ 2i\left(\gamma_1 h\bar{P}^v + \frac{\gamma_2 h}{2}\bar{P}^v\right)\sin(h\theta) + 2d + \frac{2\gamma_2\sigma_0 h^2}{2^{1-\alpha}} + \gamma_2 h^2 v\bar{P}^{v-1}\bar{P}_x\Bigg]\eta^{n+1}\\
&\quad = \Bigg[2\left(d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} - \gamma_1 h^2 v\bar{P}^{v-1}\bar{P}_x\right)\cos(h\theta)\\
&\quad - 2i\left(\gamma_1 h\bar{P}^v + \frac{\gamma_2 h}{2}\bar{P}^v\right)\sin(h\theta)\\
&\quad - 2d + \frac{2\gamma_2\sigma_0 h^2}{2^{1-\alpha}} - \gamma_2 h^2 v\bar{P}^{v-1}\bar{P}_x\Bigg]\eta^{n}
\end{aligned}$$

$$
\begin{aligned}
&- 2\sigma_0 h^2 r_1 \Big(\gamma_2 + 2\gamma_1 \cos(h\theta)\Big)\eta^n + 2\sigma_0 h^2 r_n \Big(\gamma_2 + 2\gamma_1 \cos(h\theta)\Big)\eta^0 \\
&+ 2\sigma_0 h^2 \sum_{j=1}^{n-1} \Big(r_{n-j} - r_{n-j+1}\Big)\Big(\gamma_2 + 2\gamma_1 \cos(h\theta)\Big)\eta^j .
\end{aligned} \tag{45}
$$

For $n = 0$, (45) gives

$$
\begin{aligned}
|\eta^1| &= \frac{\left|\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} - h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 - \beta_1 - 2i\beta_2 \sin(h\theta)\right|}{\left|\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 + \beta_1 + 2i\beta_2 \sin(h\theta)\right|}|\eta^0| \\
&= \frac{\sqrt{\left(\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} - h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 - \beta_1\right)^2 + 4\big(\beta_2 \sin(h\theta)\big)^2}}{\sqrt{\left(\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 + \beta_1\right)^2 + 4\big(\beta_2 \sin(h\theta)\big)^2}}|\eta^0| \\
&\leq |\eta^0|,
\end{aligned} \tag{46}
$$

where

$$
\beta_0 = \gamma_2 + 2\gamma_1 \cos(h\theta), \quad \beta_1 = 2d\left(1 - \cos(h\theta)\right), \quad \beta_2 = \gamma_1 h \bar{P}^v + \frac{\gamma_2 h}{2}\bar{P}^v. \tag{47}
$$

Now, assume that the inequality $|\eta^l| \leq |\eta^0|$ holds for $l = 2, 3, \dots, n-1$. Then, from (45), we get

$$
\begin{aligned}
|\eta^{n+1}| =& \frac{\left|\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} - h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 - \beta_1 - 2i\beta_2 \sin(h\theta) - 2\sigma_0 h^2 \beta_1 r_1\right|}{\left|\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 + \beta_1 + 2i\beta_2 \sin(h\theta)\right|}|\eta^n| \\
&+ \frac{2\sigma_0 h^2 \beta_1}{\left|\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 + \beta_1 + 2i\beta_2 \sin(h\theta)\right|} \\
&\times \left(r_n|\eta^0| + \sum_{j=1}^{n-1}(r_{n-j} - r_{n-j+1})|\eta^j|\right).
\end{aligned} \tag{48}
$$

From the assumption $|\eta^l| \leq |\eta^0|$ holds for $l = 2, 3, \dots, n-1$ and decreasing function of r_j, we have

$$
\begin{aligned}
r_n|\eta^0| + \sum_{j=1}^{n-1}(r_{n-j} - r_{n-j+1})|\eta^j| &\leq \left(\sum_{j=1}^{n-1}(r_{n-j} - r_{n-j+1}) + r_n\right)|\eta^0| \\
&= r_1|\eta^0|.
\end{aligned} \tag{49}
$$

Using (49) in (48), one obtains

$$\begin{aligned}
&|\eta^{n+1}| \\
&\leq \frac{\left|\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} - h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 - \beta_1 - 2i\beta_2 \sin(h\theta) - 2\sigma_0 h^2 \beta_1 r_1\right| + 2\sigma_0 h^2 \beta_1 r_1}{\left|\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 + \beta_1 + 2i\beta_2 \sin(h\theta)\right|} |\eta^0| \\
&\leq \frac{\left|\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} - h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 - \beta_1 - 2\sigma_0 h^2 \beta_1 r_1\right| + 2\beta_2 \sin(h\theta) + 2\sigma_0 h^2 \beta_1 r_1}{\left|\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 + \beta_1 + 2i\beta_2 \sin(h\theta)\right|} |\eta^0|, \qquad (50)
\end{aligned}$$

where β_0, β_1 and β_2 as defined in (47). As for small $\sin(\theta) \leq \theta$, we have

$$\begin{aligned}
&|\eta^{n+1}| \qquad (51) \\
&\leq \frac{\left|\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} - h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_3 - d\theta^2 h^2 - 2d\sigma_0 h^4 \theta^2 r_1\right| + 2\beta_2 h\theta + 2d\sigma_0 h^4 \theta^2 r_1}{\left|\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 + \beta_1 + 2i\beta_2 \sin(h\theta)\right|} |\eta^0|.
\end{aligned}$$

where $\beta_3 = \gamma_2 + 2\gamma_1(1 - \frac{h^2\theta^2}{2})$. Now, we consider two cases: In the first case, we consider $\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} - h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_3 - d\theta^2 h^2 > 2d\sigma_0 h^4 \theta^2 r_1$, which yields

$$\begin{aligned}
&|\eta^{n+1}| \\
&\leq \frac{\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} - h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_3 - d\theta^2 h^2 - 2d\sigma_0 h^4 \theta^2 r_1 + 2\beta_2 h\theta + 2d\sigma_0 h^4 \theta^2 r_1}{\left|\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 + \beta_1 + 2i\beta_2 \sin(h\theta)\right|} |\eta^0| \\
&\leq \frac{\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} - h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_3 - d\theta^2 h^2 + 2\beta_2 h\theta}{\left|\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 + \beta_1 + 2i\beta_2 \sin(h\theta)\right|} |\eta^0|. \qquad (52)
\end{aligned}$$

From the boundedness of $\bar{P}$ and $\bar{P}_x$, we obtain

$$\frac{2h\theta(\gamma_1 h \bar{P}^v + \frac{\gamma_2 h}{2}\bar{P}^v)}{\left|\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 + \beta_1 + 2i\beta_2 \sin(h\theta)\right|} \to 0 \text{ as } k \to 0. \qquad (53)$$

Using (53) in (52), we obtain

$$|\eta^{n+1}| \leq \frac{\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} - h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_3 - d\theta^2 h^2}{\left|\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 + \beta_1 + 2i\beta_2 \sin(h\theta)\right|} |\eta^0| \leq |\eta^0|.$$

Second, we consider $\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} - h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_3 - d\theta^2 h^2 < 2d\sigma_0 h^4 \theta^2 r_1$, which gives

$$|\eta^{n+1}| \leq \frac{4d\sigma_0 h^4 \theta^2 r_1 - \left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} - h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_3 + d\theta^2 h^2 - 2\beta_2 h\theta}{\left|\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 + \beta_1 + 2i\beta_2 \sin(h\theta)\right|} |\eta^0|. \qquad (54)$$

In order to obtain $|\eta^{n+1}| \leq |\eta^0|$ the next condition must satisfied

$$\frac{4d\sigma_0 h^4\theta^2 r_1 - \left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} - h^2 v\bar{P}^{v-1}\bar{P}_x\right)\beta_3 + d\theta^2 h^2 - 2\beta_2 h\theta}{\left|\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v\bar{P}^{v-1}\bar{P}_x\right)\beta_0 + \beta_1 + 2i\beta_2\sin(h\theta)\right|} \leq 1,$$

which is hold true when $(3/2) \leq 3^{\alpha}$. Thus, the numerical scheme in (35) is conditionally stable.

4.3 Convergence analysis

In this section, we investigate the convergence of the proposed numerical scheme (35), for which the error is defined as $\xi_m^n = p_m^n - P_m^n, m = 0, 1, 2, \ldots, N_x, n = 0, 1, 2, \ldots, M_t$, at the grid point (x_m, t_n). Then, the error equation is written in the following form:

$$\begin{aligned}
&\left[-d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} + \gamma_1 h^2 v\bar{P}^{v-1}\bar{P}_x - \gamma_1 h\bar{P}^v - \frac{\gamma_2 h}{2}\bar{P}^v\right]\xi_{m-1}^{n+1}\\
&+\left[2d + \frac{2\gamma_2\sigma_0 h^2}{2^{1-\alpha}} + \gamma_2 h^2 v\bar{P}^{v-1}\bar{P}_x\right]\xi_m^{n+1}\\
&+\left[-d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} + \gamma_1 h^2 v\bar{P}^{v-1}\bar{P}_x + \gamma_1 h\bar{P}^v + \frac{\gamma_2 h}{2}\bar{P}^v\right]\xi_{m+1}^{n+1}\\
&\quad=\left[d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} - \gamma_1 h^2 v\bar{P}^{v-1}\bar{P}_x + \gamma_1 h\bar{P}^v + \frac{\gamma_2 h}{2}\bar{P}^v\right]\xi_{m-1}^{n}\\
&\qquad+\left[-2d + \frac{2\gamma_2\sigma_0 h^2}{2^{1-\alpha}} - \gamma_2 h^2 v\bar{P}^{v-1}\bar{P}_x\right]\xi_m^{n}\\
&\qquad+\left[d + \frac{2\gamma_1\sigma_0 h^2}{2^{1-\alpha}} - \gamma_1 h^2 v\bar{P}^{v-1}\bar{P}_x - \gamma_1 h\bar{P}^v - \frac{\gamma_2 h}{2}\bar{P}^v\right]\xi_{m+1}^{n}\\
&\qquad- 2\sigma_0 h^2 r_1\left(\gamma_1\xi_{m-1}^n + \gamma_2\xi_m^n + \gamma_1\xi_{m+1}^n\right)\\
&\qquad+ 2\sigma_0 h^2 r_n\left(\gamma_1\xi_{m-1}^0 + \gamma_2\xi_m^0 + \gamma_1\xi_{m+1}^0\right)\\
&\qquad- 2\sigma_0 h^2\sum_{i}^{n-1}\left(r_{n-i+1} - r_{n-i}\right)\left(\gamma_1\xi_{m-1}^i + \gamma_2\xi_m^i + \gamma_1\xi_{m+1}^i\right)\\
&\qquad+ 2h^2\left(\gamma_1\mathcal{R}_{m-1}^{n+\frac{1}{2}} + \gamma_2\mathcal{R}_m^{n+\frac{1}{2}} + \gamma_1\mathcal{R}_{m+1}^{n+\frac{1}{2}}\right),
\end{aligned} \tag{55}$$

associated with the required conditions in terms of error term $\xi_0^n = \xi_{N_x}^n = \xi_m^0 = 0$ and $\mathcal{R}_m^{n+\frac{1}{2}} \leq C\mathcal{T}_m^{n+\frac{1}{2}} \leq C^1\left(h^2 + k^{2-\alpha}\right)$. Likewise, the grid functions defined in the previous section, we use the following grid functions in the further process of proving the convergence:

$$\xi^n(x) = \begin{cases} \xi_m^n, & x_{m-\frac{h}{2}} \leq x \leq x_{m+\frac{h}{2}}, \quad 1 \leq m \leq N_x - 1,\\ 0, & 0 \leq x \leq \frac{h}{2} \text{ or } x_{L-\frac{h}{2}} \leq x \leq L, \end{cases}$$

and

$$\mathcal{R}^{n+\frac{1}{2}}(x) = \begin{cases} \mathcal{R}_m^{n+\frac{1}{2}}, & x_{m-\frac{h}{2}} \le x \le x_{m+\frac{h}{2}}, \quad 1 \le m \le N_x - 1, \\ 0, & 0 \le x \le \frac{h}{2} \text{ or } x_{L-\frac{h}{2}} \le x \le L. \end{cases}$$

Expanding the grid functions $\mathcal{R}^{n+\frac{1}{2}}(x)$ and $\xi^n(x)$ using the Fourier series, we have

$$\xi^n(x) = \sum_{l=-\infty}^{\infty} \delta^n(l) e^{i\frac{2\pi l x}{L}},$$
$$\mathcal{R}^{n+\frac{1}{2}}(x) = \sum_{l=-\infty}^{\infty} r^{n+\frac{1}{2}}(l) e^{i\frac{2\pi l x}{L}}, \quad n = 0, 1, 2, \ldots, M_t,$$

where $\delta^n(l)$ and $r^{n+\frac{1}{2}}(l)$ are defined as

$$\delta^n(l) = \frac{1}{L}\int_0^L \xi^n(s) e^{-i\frac{2\pi l s}{L}} ds, \quad r^{n+\frac{1}{2}}(l) = \frac{1}{L}\int_0^L \mathcal{R}^{n+\frac{1}{2}}(s) e^{-i\frac{2\pi l s}{L}} ds.$$

Using Parseval's identity and the L_2-norm, we get

$$\|\xi^n\|_2^2 = \sum_{n=1}^{M_t-1} h|\xi_m^n|^2 = \int_0^L |\xi^n(s)|^2 ds = L \sum_{l=-\infty}^{\infty} |\delta^n(l)|^2, \tag{56}$$

and

$$\|\mathcal{R}^{n+\frac{1}{2}}\|_2^2 = \sum_{n=1}^{M_t-1} h|\mathcal{R}_m^{n+\frac{1}{2}}|^2 = \int_0^L |\mathcal{R}^{n+\frac{1}{2}}(s)|^2 ds = L \sum_{l=-\infty}^{\infty} |r^{n+\frac{1}{2}}(l)|^2. \tag{57}$$

Assuming that $\xi_m^n = \delta^n e^{ihm\theta}$ and $\mathcal{R}_m^{n+\frac{1}{2}} = r^{n+\frac{1}{2}} e^{ihm\theta}$, (55) gives

$$\begin{aligned} \delta^{n+1} =& \frac{\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} - h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 - \beta_1 - 2i\beta_2 \sin(h\theta) - 2\sigma_0 h^2 r_1(\gamma_2 + 2\gamma_1\cos(h\theta))}{\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 + \beta_1 + 2i\beta_2\sin(h\theta)}\delta^n \\ &+ \frac{2\sigma_0 h^2 \beta_0 r_n}{\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 + \beta_1 + 2i\beta_2\sin(h\theta)}\delta^0 \\ &+ \frac{2\sigma_0\beta_1 h^2 \sum_{j=1}^{n-1}(r_{n-j} - r_{n-j+1})}{\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 + \beta_1 + 2i\beta_2\sin(h\theta)}\delta^j \\ &+ \frac{2h^2\beta_0}{\left(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v \bar{P}^{v-1}\bar{P}_x\right)\beta_0 + \beta_1 + 2i\beta_2\sin(h\theta)} r^{n+\frac{1}{2}}. \end{aligned} \tag{58}$$

Lemma 1. If δ^n satisfy (58), then there exists a positive constant C such that

$$|\delta^n| \le C|r^{\frac{1}{2}}|, \qquad n = 0, 1, 2, \ldots, M_t.$$

Proof. We will prove the result using mathematical induction. Since the error equation satisfies the initial condition $\xi_m^0 = 0$, it follows that $r^0 = 0$. Thus, for $n = 0$, (58) gives

$$|\delta^1| = \frac{2h^2\beta_0}{|\big(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v\bar{P}^{v-1}\bar{P}_x\big)\beta_0 + \beta_1 + 2i\beta_2\sin(h\theta)|}|r^{\frac{1}{2}}|.$$

It implies that $|\delta^1| \leq C|r^{\frac{1}{2}}|$, for some $C \leq 1$. Now, we assume that the result is true for n such that $|\delta^n| \leq C|r^{\frac{1}{2}}||$. From (58), we get

$$\begin{aligned}
|\delta^{n+1}| \leq & \frac{\left|\big(\frac{2\sigma_0 h^2}{2^{1-\alpha}} - h^2 v\bar{P}^{v-1}\bar{P}_x\big)\beta_0 - \beta_1 - 2i\beta_2\sin(h\theta) - 2\sigma_0 h^2\beta_0 r_1\right|}{\left|\big(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v\bar{P}^{v-1}\bar{P}_x\big)\beta_0 + \beta_1 + 2i\beta_2\sin(h\theta)\right|}|\delta^n| \\
& + \frac{|2h^2\beta_0|\left(\sigma_0\sum_{j=1}^{n-1}(r_{n-j} - r_{n-j+1})|\delta^j| + \sigma_0 r^n|\delta^0| + |r^{n+\frac{1}{2}}|\right)}{\left|\big(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v\bar{P}^{v-1}\bar{P}_x\big)\beta_0 + \beta_1 + 2i\beta_2\sin(h\theta)\right|} \\
\leq & \frac{\left|\big(\frac{2\sigma_0 h^2}{2^{1-\alpha}} - h^2 v\bar{P}^{v-1}\bar{P}_x\big)\beta_0 - \beta_1 - 2i\beta_2\sin(h\theta) - 2\sigma_0 h^2\beta_0 r_1\right|}{\left|\big(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v\bar{P}^{v-1}\bar{P}_x\big)\beta_0 + \beta_1 + 2i\beta_2\sin(h\theta)\right|}C^1|r^{\frac{1}{2}}| \\
& + \frac{|2\sigma_0\beta_0 h^2 r_1|}{\left|\big(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v\bar{P}^{v-1}\bar{P}_x\big)\beta_0 + \beta_1 + 2i\beta_2\sin(h\theta)\right|}C^2|r^{\frac{1}{2}}| \\
& + \frac{|2h^2\beta_0|}{\left|\big(\frac{2\sigma_0 h^2}{2^{1-\alpha}} + h^2 v\bar{P}^{v-1}\bar{P}_x\big)\beta_0 + \beta_1 + 2i\beta_2\sin(h\theta)\right|}|r^{n+\frac{1}{2}}|. \qquad (59)
\end{aligned}$$

Using (41) and (57), we get that there exists a positive constant $C^{n+\frac{1}{2}}$, such that

$$|r^{n+\frac{1}{2}}| \leq C^{n+\frac{1}{2}}|r^{\frac{1}{2}}|, \qquad n = 0, 1, 2, \ldots, M_t.$$

Let $C = \max\{C^1, C^2, C^{n+\frac{1}{2}}, n = 0, 1, 2, \ldots, M_t\}$, and let the time step size approach zero. Then we obtain $|\delta^n| \leq C|r^{\frac{1}{2}}|$. □

Theorem 1. The proposed numerical scheme (35) is convergent, and the solution satisfies the condition $\|p^n - P^n\| \leq C(h^2 + k^{2-\alpha}),\ n = 0, 1, 2, \ldots, M_t$.

Proof. Using the result of Lemma 1 and the Parseval's identity presented in (56) and (57), we get the following result:

$$\|p^n - P^n\| = \|\xi^n\|_2 \leq C\|\mathcal{R}^{\frac{1}{2}}\|_2 = C(h^2 + k^{2-\alpha}).$$

Hence, the convergence order relies on the appropriate choice of parameters γ_1 and γ_2. When $\gamma_1 = \frac{1}{12}$ and $2\gamma_1 + \gamma_2 = 1$, the proposed numerical scheme is $O\left(h^2 + k^{2-\alpha}\right)$ accurate. $\square$

Remark 2. Although the nonlinear parameter v can be significantly affect the qualitative behavior of continuous solution, the numerical scheme's convergence orders and stability condition are independent of v. This is because the quasi-linearization process places v only within bounded coefficients, and the error analysis relies solely on the boundedness of these coefficients rather than their specific values. The numerical results recorded in tables and figures confirm that for $v = 1, 2, \ldots, 10$, the errors and convergence rates remain comparable.

5 Numerical simulation

In this section, some numerical experiments are carried out to validate the theoretical results. Let $p(x_m, t_n)$ and $P(x_m, t_n)$ denote the analytical and numerical solutions to the problem (1) at the grid point (x_m, t_n). The errors are measured using the following error norms:

$$E_2^{N_x,M_t} = \max_{1\le n\le M_t} \sqrt{h \sum_{m=1}^{N_x-1} \left|p(x_m,t_n) - P(x_m,t_n)\right|^2},$$
$$E_\infty^{N_x,M_t} = \max_{1\le n\le M_t} \left(\max_{1\le m\le N_x-1} \left|p(x_m,t_n) - P(x_m,t_n)\right| \right).$$

For problems where the exact solution is not available, the efficiency of the proposed method is evaluated by computing the errors in the L_2 and L_∞-norms using the double mesh principle, defined as follows:

$$E_2^{N_x,M_t} = \max_{1\le n\le M_t} \sqrt{h \sum_{m=1}^{N_x-1} \left|P(x_m,t_n) - \bar{P}(x_{2m},t_{2n})\right|^2}, \tag{60}$$
$$E_\infty^{N_x,M_t} = \max_{1\le n\le M_t} \left(\max_{1\le m\le N_x-1} \left|P(x_m,t_n) - \bar{P}(x_{2m},t_{2n})\right| \right), \tag{61}$$

where $\bar{P}(x_{2m}, t_{2n})$, for $m = 0, 1, 2, \ldots, N_x, n = 0, 1, 2, \ldots, M_t$, denotes the numerical solution obtained on a refined grid by doubling the number of mesh points in both spatial and temporal directions. The spatial and temporal convergence orders of the method are computed using the following formulas:

$$L_q^x = \log_2 \left(\frac{E_q^{N_x,M_t}}{E_q^{2N_x,M_t}} \right), \text{ where } q = 2, \infty, \tag{62}$$

$$L_q^t = \log_2 \left(\frac{E_q^{N_x,M_t}}{E_q^{N_x,2M_t}} \right), \text{ where } q = 2, \infty. \tag{63}$$

Furthermore, we investigate the combined spatial and temporal convergence orders of the method using the formula:

$$\rho^{N_x,M_t} = \log_2 \left(\frac{E_q^{N_x,M_t}}{E_q^{2N_x,2M_t}} \right), \text{ where } q = 2, \infty. \tag{64}$$

Based on the structure of the proposed method, a quasi-linearized system of equations is solved at each time level. In this study, the proposed numerical scheme is combined with the Thomas algorithm as an iterative solver to obtain the numerical solution. In practice, the stopping criteria is defined as $\|p_m^n - P_m^n\| \leq 10^{-8}$, where p_m^n denotes the exact solution and P_m^n denotes the numerical solution. For all numerical implementations in this paper, we use the parameter values $\gamma_1 = \frac{1}{12}$ and $\gamma_2 = \frac{10}{12}$.

Example 1. Consider the following generalized nonlinear time-fractional convection-diffusion equation [13]:

$${}_0^C D_t^\alpha p(x,t) + (p(x,t))^v \frac{\partial p(x,t)}{\partial x} = d \frac{\partial^2 p(x,t)}{\partial x^2} + f(x,t), (x,t) \in (0,1) \times (0,1),$$

with initial and boundary conditions

$$\begin{aligned} p(x,0) &= 0, \quad 0 \leq x \leq 1, \\ p(0,t) &= t^2, \quad p(1,t) = et^2, \quad t \geq 0. \end{aligned}$$

The source term is defined as $f(x,t) = \frac{2t^{2-\alpha} e^x}{\Gamma(3-\alpha)} + (t^2 e^x)^{v+1} - dt^2 e^x$. For special value of $\alpha = 1$, the exact solution is $p(x,t) = e^x t^2$.

For this example, the numerical experiments are conducted for various values of α, v, and the mesh sizes, and the corresponding results are presented in tables and figures. Since the exact solution is available only for $\alpha = 1$, the double mesh principle is employed for other values of α. Tables 1–4 present a comprehensive numerical study of the proposed scheme with respect to temporal and spatial discretizations, as well as the influence of the nonlinear convection parameter v and the fractional order α. Table 1 reports the temporal errors, convergence orders, and CPU times of the computed solution for different values of α with a sufficiently fine spatial grid $N_x = 1000$ using formula in (63), ensuring that spatial errors do not influence temporal accuracy. It is observed that the errors in both L_2^t and L_∞^t-norms decrease monotonically as the time step size is refined, confirming the stability and convergence of the temporal discretization. The computed convergence rates are in good agreement with the theoretical order $2-\alpha$. In particular, for a smaller fractional order $\alpha = 0.2$, the theoretical rate is $2-\alpha = 1.8$, and observe rates are initially close to this value and then slightly decrease toward 1.8 as the mesh is further refined, indicating convergence toward the theoretical estimate. A similar trend is observed for other values of α: for $\alpha = 0.5$, the rates approach the expected order 1.5, while for $\alpha = 0.5$ and $\alpha = 0.9$, the computed rates consistently align with the theoretical orders 1.3 and 1.1, respectively.

Regarding the stability condition stated in Remark 3, namely $3^{\alpha} \geq \frac{3}{2}$ (implying $\alpha \geq 0.37$), it is important to emphasize that this is a sufficient not necessary condition. The numerical results clearly show that the scheme remains stable and accurate even for $\alpha = 0.2$, where the condition is violated. This confirms that the theoretical stability bound is conservative, as commonly observed in von Neumann-type analyses and energy-based estimates. The inclusion of $\alpha = 0.2$ is therefore deliberate, demonstrating the robustness of the method beyond the theoretically guarantee range.

Table 1: Temporal errors, convergence orders, and CPU times of the computed solution for Example 1 with different fractional orders α and time step sizes M_t, using $N_x = 1000$ and $d = v = 1$.

α	Norms	$M_t = 32$	$M_t = 64$	$M_t = 128$	$M_t = 256$	$M_t = 512$	$M_t = 1024$
	L_2^t	1.14×10^{-4}	3.01×10^{-5}	7.81×10^{-6}	2.02×10^{-6}	5.79×10^{-7}	1.66×10^{-7}
		1.928	1.945	1.949	1.805	1.798	—
0.2	L_∞^t	1.60×10^{-4}	4.21×10^{-5}	1.09×10^{-5}	2.82×10^{-6}	7.87×10^{-7}	2.26×10^{-7}
		1.924	1.947	1.953	1.843	1.800	—
	CUP(s)	3.3953	6.9406	17.5882	41.9596	128.6501	420.4133
	L_2^t	1.98×10^{-4}	6.21×10^{-5}	1.85×10^{-5}	5.55×10^{-6}	1.82×10^{-6}	6.03×10^{-7}
		1.676	1.744	1.740	1.609	1.593	—
0.5	L_∞^t	2.67×10^{-4}	8.33×10^{-5}	2.47×10^{-5}	7.65×10^{-6}	2.50×10^{-6}	8.28×10^{-7}
		1.684	1.752	1.692	1.6114	1.598	—
	CUP(s)	0.5482	2.8717	6.5998	5.9386	32.0136	66.3795
	L_2^t	2.53×10^{-4}	9.11×10^{-5}	3.40×10^{-5}	1.31×10^{-5}	5.26×10^{-6}	2.12×10^{-6}
		1.475	1.419	1.373	1.321	1.306	—
0.7	L_∞^t	3.50×10^{-4}	1.25×10^{-4}	4.69×10^{-5}	1.80×10^{-5}	7.21×10^{-6}	2.91×10^{-6}
		1.478	1.421	1.378	1.324	1.307	—
	CUP(s)	0.1335	0.2927	0.6866	1.7674	4.3070	14.4766
	L_2^t	2.43×10^{-4}	9.86×10^{-5}	4.23×10^{-5}	1.92×10^{-5}	8.98×10^{-6}	4.19×10^{-6}
		1.300	1.220	1.136	1.102	1.099	—
0.9	L_∞^t	3.35×10^{-4}	1.36×10^{-4}	5.83×10^{-5}	2.64×10^{-5}	1.23×10^{-6}	5.74×10^{-6}
		1.304	1.222	1.142	1.102	1.099	—
	CUP(s)	0.1335	0.2927	0.6866	1.7674	4.3070	14.4766

Table 2: Spatial errors, convergence orders, and CPU times of the computed solution for Example 1 with different values of α and the space step sizes N_x and $M_t = 1000$ with $d = v = 1$.

α	Norms	$N_x = 4$	$N_x = 8$	$N_x = 16$	$N_x = 32$	$N_x = 64$
	L_2^x	1.13×10^{-3}	2.73×10^{-4}	6.76×10^{-5}	1.67×10^{-5}	4.07×10^{-6}
		2.047	2.017	2.013	2.039	—
0.2	L_∞^x	1.53×10^{-3}	3.82×10^{-4}	9.45×10^{-5}	2.35×10^{-5}	5.72×10^{-6}
		2.005	2.016	2.008	2.039	—
	CUP(s)	17.2782	14.5134	14.3909	14.6191	12.0917
	L_2^x	1.09×10^{-3}	2.65×10^{-4}	6.53×10^{-5}	1.60×10^{-5}	3.74×10^{-6}
		2.047	2.020	2.027	2.099	—
0.4	L_∞^x	1.49×10^{-3}	3.70×10^{-4}	9.14×10^{-5}	2.25×10^{-5}	5.25×10^{-6}
		2.010	2.017	2.022	2.098	—
	CUP(s)	17.0886	14.4452	14.3717	14.4228	11.8596
	L_2^x	1.02×10^{-3}	2.47×10^{-4}	5.96×10^{-5}	1.33×10^{-5}	2.12×10^{-6}
		2.053	2.052	2.165	2.648	—
0.7	L_∞^x	1.40×10^{-3}	3.44×10^{-4}	8.36×10^{-5}	1.87×10^{-5}	2.90×10^{-6}
		2.024	2.044	2.158	2.689	—
	CUP(s)	17.0548	14.6582	14.3671	14.3693	11.8885

Table 2 presents the spatial discretization errors and CPU times of the computed solution using a sufficiently fine time step $M_t = 1000$, ensuring that the temporal errors do not influence spatial accuracy. The results clearly demonstrate that the method achieves second-order accuracy in space for all considered values of α. The observed convergence rates remain uniformly close to 2 across all refinement levels, confirming that the spatial discretization is independent of the fractional-order α and is not affected by temporal memory effects.

Tables 3 and 4 examine the effect of the nonlinear parameter v in the term $p^v p_x$. It is observed Table 3 that the errors in both L_2 and L_∞-norms decrease consistently with mesh refinement, and the corresponding convergence rates remain stable for all values $v = 1, 2, \ldots, 10$. This indicates that the proposed scheme maintains its accuracy even for strongly nonlinear cases. Moreover, under further mesh refinement, and as also observed in Table 1, the numerical convergence orders approach the expected theoretical values more closely. Table 4 reports pointwise errors for fixed discretization parameters $N_x = 160$ and $M_t = 256$, for various values of α and v. The results indicates that the numerical errors remain uniformly small across all tested values of α and v, demonstrating the stability and robustness of the scheme. As α increases, a slight improvement in accuracy is observed, which is consistent with the smoother behavior of solutions for larger fractional orders. In line with Remark 2, the convergence behavior and stability of the scheme are not significantly influenced by v. This is reflected in the numerical results, where the errors and convergence rates remain comparable across all tested values of v, demonstrating

Table 3: Pointwise errors and corresponding convergence orders for Example 1 in L_2 and L_∞ norms for different values of v with $\alpha = 0.5$ and $d = 1$.

v	Norms	$N_x = 10$, $M_t = 16$	$N_x = 20$, $M_t = 32$	$N_x = 40$, $M_t = 64$	$N_x = 80$, $M_t = 128$	$N_x = 160$, $M_t = 256$	$N_x = 320$, $M_t = 512$
1	L_2	1.533×10^{-3}	5.262×10^{-4}	1.684×10^{-4}	5.110×10^{-5}	1.487×10^{-5}	4.186×10^{-6}
		1.54	1.64	1.72	1.78	1.83	
	L_∞	2.069×10^{-3}	7.071×10^{-4}	2.254×10^{-4}	6.800×10^{-5}	1.968×10^{-5}	5.515×10^{-6}
		1.55	1.65	1.73	1.79	1.84	—
2	L_2	1.522×10^{-3}	5.256×10^{-4}	1.684×10^{-4}	5.110×10^{-5}	1.487×10^{-5}	4.186×10^{-6}
		1.53	1.64	1.72	1.78	1.83	—
	L_∞	2.053×10^{-3}	7.063×10^{-4}	2.253×10^{-4}	6.800×10^{-5}	1.968×10^{-5}	5.515×10^{-6}
		1.54	1.65	1.73	1.79	1.84	—
3	L_2	5.567×10^{-3}	1.219×10^{-3}	2.827×10^{-4}	6.651×10^{-5}	1.542×10^{-5}	4.186×10^{-6}
		2.19	2.11	2.09	2.11	1.88	—
	L_∞	9.333×10^{-3}	1.958×10^{-3}	4.482×10^{-4}	1.063×10^{-4}	2.519×10^{-5}	5.842×10^{-6}
		2.25	2.13	2.08	2.08	2.11	—
4	L_2	1.289×10^{-2}	2.635×10^{-3}	6.058×10^{-4}	1.457×10^{-4}	3.551×10^{-5}	8.629×10^{-6}
		2.29	2.12	2.06	2.04	2.04	—
	L_∞	2.445×10^{-2}	5.568×10^{-3}	1.142×10^{-3}	2.415×10^{-4}	5.901×10^{-5}	1.450×10^{-5}
		2.13	2.29	2.24	2.03	2.02	—
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
10	L_2	4.903×10^{-2}	1.245×10^{-2}	2.924×10^{-3}	7.070×10^{-4}	1.696×10^{-4}	4.004×10^{-5}
		1.98	2.09	2.05	2.06	2.08	—
	L_∞	9.751×10^{-2}	2.833×10^{-2}	8.120×10^{-3}	2.208×10^{-3}	5.768×10^{-4}	1.460×10^{-4}
		1.78	1.80	1.88	1.94	1.98	—

Table 4: Pointwise errors for Example 1 in L_2 and L_∞ norms for different values of α and v using $N_x = 160, M_t = 256$ and $d = 1$.

v	Norms	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$	$\alpha = 0.9$	$\alpha = 1$
1	L_2	1.367×10^{-5}	1.421×10^{-5}	2.402×10^{-5}	2.219×10^{-5}	4.760×10^{-7}
	L_∞	1.830×10^{-5}	1.941×10^{-5}	3.294×10^{-5}	3.045×10^{-5}	6.528×10^{-7}
2	L_2	1.367×10^{-5}	1.421×10^{-5}	2.325×10^{-5}	2.137×10^{-5}	3.834×10^{-6}
	L_∞	1.830×10^{-5}	1.866×10^{-5}	3.178×10^{-5}	2.921×10^{-5}	5.495×10^{-6}
3	L_2	1.714×10^{-5}	1.421×10^{-5}	2.306×10^{-5}	2.116×10^{-5}	1.792×10^{-5}
	L_∞	2.714×10^{-5}	2.248×10^{-5}	3.155×10^{-5}	2.895×10^{-5}	2.776×10^{-5}
4	L_2	3.654×10^{-5}	3.409×10^{-5}	3.086×10^{-5}	3.104×10^{-5}	3.680×10^{-5}
	L_∞	5.992×10^{-5}	5.778×10^{-5}	5.496×10^{-5}	5.513×10^{-5}	6.004×10^{-5}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
10	L_2	1.697×10^{-4}	1.695×10^{-4}	1.692×10^{-4}	1.692×10^{-4}	1.695×10^{-4}
	L_∞	5.768×10^{-4}	5.768×10^{-4}	5.767×10^{-4}	5.767×10^{-4}	5.767×10^{-4}

the robustness of the method with respect to nonlinear effects. A numerical comparison of the suggested method with the current numerical approaches documented in [13], [21], [22], and [35] is presented in Tables 5–7. Comparing the suggested approach to the previously published schemes, the numerical results show that it achieves better convergence characteristics and higher accuracy. Results specifically show that the current approach is robust and efficient in all tested problem settings, confirming its efficacy and potential benefits for real-world applications across a broad range of v values.

Table 5: Comparison results for Example 1 in the L_2 and L_∞ norms for $\alpha = 0.4, d = 1$, and different values of v and mesh sizes.

v	Norms	$N_x = 16, M_t = 10$		$N_x = 32, M_t = 20$		$N_x = 64, M_t = 40$	
		Method [13]	Present	Method [13]	Present	Method [13]	Present
2	L_2	1.08×10^{-2}	6.314×10^{-3}	2.53×10^{-3}	1.338×10^{-3}	6.15×10^{-4}	3.614×10^{-4}
	L_∞	1.73×10^{-2}	8.962×10^{-3}	4.09×10^{-3}	1.917×10^{-3}	9.92×10^{-4}	5.141×10^{-4}
3	L_2	1.61×10^{-2}	8.561×10^{-3}	3.75×10^{-3}	1.611×10^{-3}	9.15×10^{-4}	4.148×10^{-4}
	L_∞	3.56×10^{-2}	1.263×10^{-2}	8.64×10^{-3}	2.482×10^{-3}	2.10×10^{-3}	6.167×10^{-4}

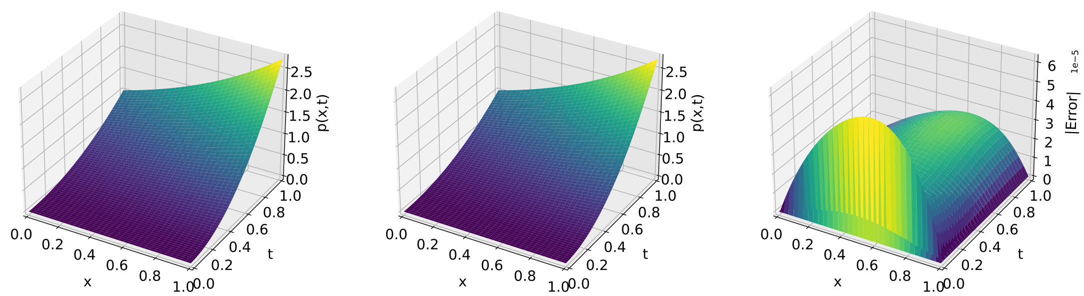
Table 6: L_∞ and L_2 errors comparisons for Example 1 with $M_t = 4000$, $\alpha = 0.5$ and $v = 1$.

N_x	Method [21]		Method [22]		Present method	
	L_2	L_∞	L_2	L_∞	L_2	L_∞
10	1.633×10^{-3}	2.296×10^{-3}	1.765×10^{-3}	3.101×10^{-3}	5.136×10^{-4}	7.113×10^{-4}
20	4.477×10^{-4}	6.250×10^{-4}	4.056×10^{-4}	8.128×10^{-4}	1.246×10^{-4}	1.737×10^{-4}
40	1.618×10^{-4}	2.273×10^{-4}	6.774×10^{-5}	2.095×10^{-4}	3.081×10^{-5}	4.320×10^{-5}
80	9.262×10^{-5}	1.331×10^{-4}	4.575×10^{-5}	6.920×10^{-5}	7.580×10^{-6}	1.063×10^{-5}

Table 7: L_∞ and L_2 errors comparisons for Example 1 with $\alpha = 0.5, v = 1$ and $N_x = 40$.

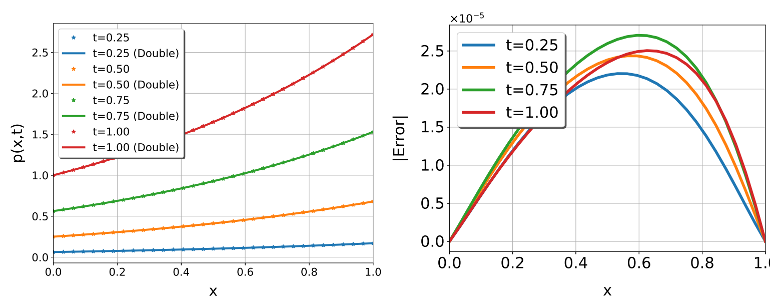
M_t	Method [35]		Present method	
	L_2	L_∞	L_2	L_∞
1000	9.204×10^{-4}	2.401×10^{-3}	7.474×10^{-6}	1.089×10^{-5}
2000	4.549×10^{-4}	1.198×10^{-3}	6.639×10^{-6}	9.759×10^{-6}
4000	2.225×10^{-4}	5.972×10^{-4}	6.363×10^{-6}	9.385×10^{-6}
8000	8.344×10^{-5}	2.364×10^{-4}	6.271×10^{-6}	9.259×10^{-6}

Figure 1 displays the numerical solution and absolute error surface plots, while Figure 2 displays the two-dimensional (2D) numerical solution and error plots for Example 1 at $\alpha = 0.4, d = v = 1$. Graphs of absolute errors for various values of α and v at various time levels $t = [0.25, 0.5, 0.75, 1.0]$ are shown in Figures 3–8. All errors in Figures 2–8 are computed using the double mesh principle, based on the difference between the numerical solutions obtained on the meshes $(N_x, M_t) = (64, 128)$ and $(N_x, M_t) = (128, 256)$ for Example 1.



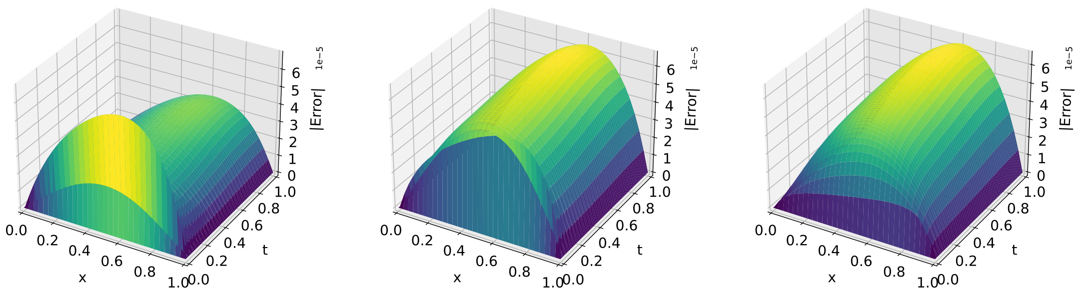
(a) $N_x = 64, M_t = 128$ (b) $N_x = 128, M_t = 256$ (c) Error

Figure 1: 3D representations of the solution behavior and corresponding absolute error for Example 1 with $\alpha = 0.4, v = 1$ and $d = 1$.



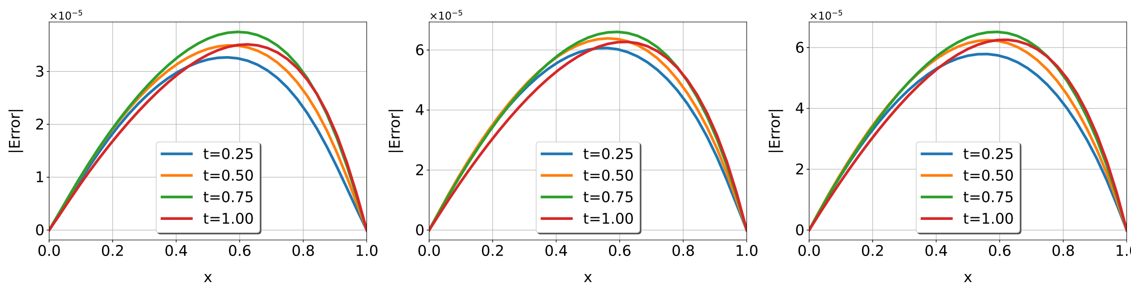
(a) Solution comparison (b) Error

Figure 2: 2D representations of the solution comparison and the corresponding absolute error for Example 1 with $\alpha = 0.4, v = 1$ and $d = 1$.



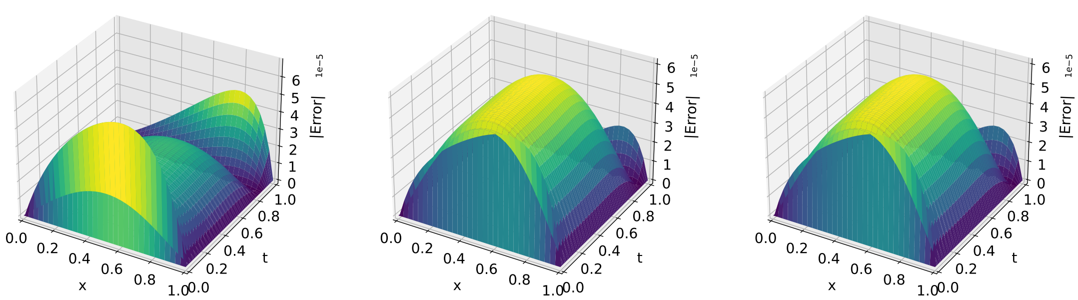
(a) Error for $\alpha = 0.5$ (b) Error for $\alpha = 0.7$ (c) Error for $\alpha = 0.9$

Figure 3: 3D representations of the absolute error behaviors for Example 1 with $v = 1$ and $d = 1$.



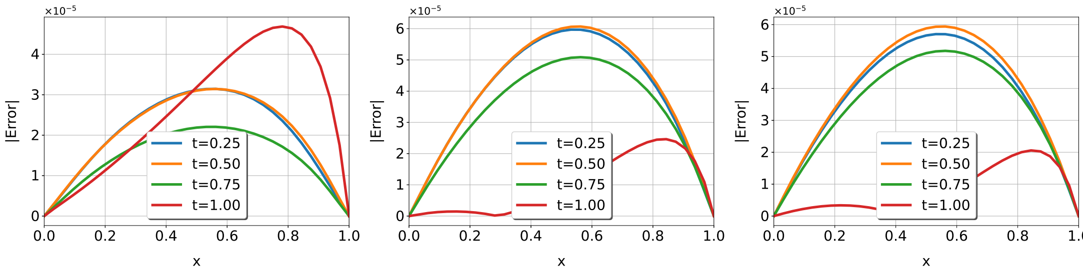
(a) Error for $\alpha = 0.5$ (b) Error for $\alpha = 0.7$ (c) Error for $\alpha = 0.9$

Figure 4: 2D representations of the absolute error behaviors for Example 1 at $v = d = 1$.



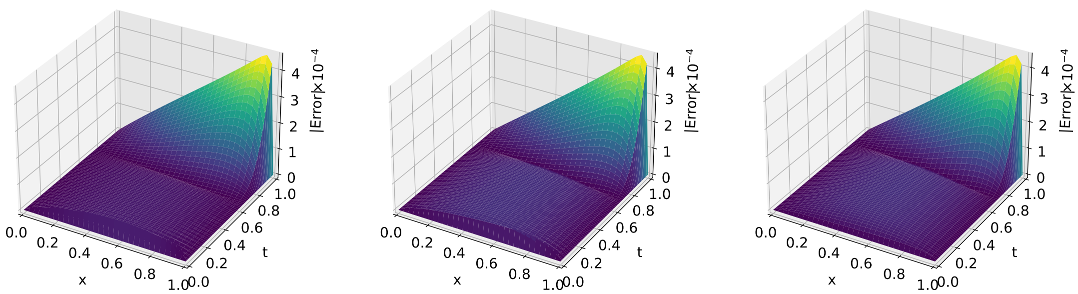
(a) Error for $\alpha = 0.5$ (b) Error for $\alpha = 0.7$ (c) Error for $\alpha = 0.9$

Figure 5: 3D representations of the absolute error behaviors for Example 1 at $v = 2, d = 1$.



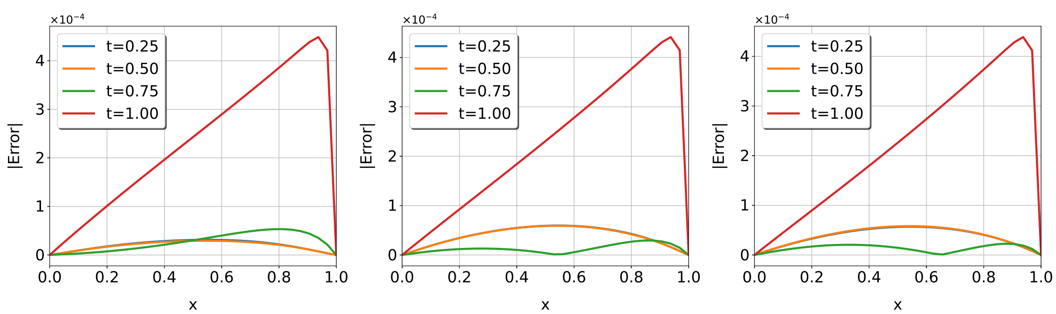
(a) Error for $\alpha = 0.5$ (b) Error for $\alpha = 0.7$ (c) Error for $\alpha = 0.9$

Figure 6: 2D representations of the absolute error behaviors for Example 1 at $v = 2, d = 1$.



(a) Error for $\alpha = 0.5$ (b) Error for $\alpha = 0.7$ (c) Error for $\alpha = 0.9$

Figure 7: 3D representations of the absolute error behaviors for Example 1 at $v = 4, d = 1$.



(a) Error for $\alpha = 0.5$ (b) Error for $\alpha = 0.7$ (c) Error for $\alpha = 0.9$

Figure 8: 2D representations of the absolute error behaviors for Example 1 at $v = 4, d = 1$.

Example 2. Consider the following generalized nonlinear time-fractional convection-diffusion equation [13]:

$$ {}_0^C D_t^\alpha p(x,t) + (p(x,t))^v \frac{\partial p(x,t)}{\partial x} = d\frac{\partial^2 p(x,t)}{\partial x^2} + f(x,t), (x,t) \in (0,1) \times (0,1), $$

with initial and boundary conditions

$$ \begin{aligned} p(x,0) &= 0, \quad 0 \le x \le 1, \\ p(0,t) &= 0, \quad p(1,t) = 0, \quad t \ge 0, \end{aligned} $$

where the source term is $f(x,t) = \frac{2t^{2-\alpha}\sin(2\pi x)}{\Gamma(3-\alpha)} + 2\pi t^4 \sin(2\pi x)\cos(2\pi x) + 4dt^2\pi^2 \sin(2\pi x)$, and $p(x,t) = t^2 \sin(2\pi x)$ is the exact solution of this problem for special case $\alpha = 1$ and $v = 1$ [35, 25].

Table 8: Pointwise errors and corresponding convergence orders for Example 2 in the L_2 and L_∞ norms for $\alpha = 0.5, d = 1$, and different values of v and mesh sizes.

v	Norms	$N_x = 10$ $M_t = 16$	$N_x = 20$ $M_t = 32$	$N_x = 40$ $M_t = 64$	$N_x = 80$ $M_t = 128$	$N_x = 160$ $M_t = 256$	$N_x = 320$ $M_t = 512$
1	L_2	6.03×10^{-3}	1.24×10^{-3}	3.02×10^{-4}	7.24×10^{-5}	1.72×10^{-5}	4.08×10^{-6}
		2.27	2.05	2.06	2.07	2.08	—
	L_∞	8.65×10^{-3}	1.67×10^{-3}	4.27×10^{-4}	1.02×10^{-4}	2.44×10^{-5}	5.77×10^{-6}
		2.37	1.97	2.06	2.07	2.08	—
2	L_2	1.66×10^{-2}	3.87×10^{-3}	1.01×10^{-3}	2.59×10^{-4}	6.61×10^{-5}	1.68×10^{-5}
		2.10	1.93	1.97	1.98	1.97	—
	L_∞	2.53×10^{-2}	6.56×10^{-3}	1.71×10^{-3}	4.36×10^{-4}	1.11×10^{-4}	2.82×10^{-5}
		1.95	1.94	1.97	1.98	1.97	—
3	L_2	5.07×10^{-3}	1.24×10^{-3}	3.02×10^{-4}	7.24×10^{-5}	1.72×10^{-5}	4.08×10^{-6}
		2.02	2.05	2.06	2.07	2.08	—
	L_∞	6.83×10^{-3}	1.67×10^{-3}	4.27×10^{-4}	1.02×10^{-4}	2.44×10^{-5}	5.77×10^{-6}
		2.02	1.97	2.06	2.07	2.08	—
4	L_2	1.95×10^{-2}	2.80×10^{-3}	8.02×10^{-4}	2.10×10^{-4}	5.41×10^{-5}	1.39×10^{-5}
		2.80	1.80	1.93	1.96	1.96	—
	L_∞	2.61×10^{-2}	4.59×10^{-3}	1.33×10^{-3}	3.51×10^{-4}	9.03×10^{-5}	2.32×10^{-5}
		2.51	1.78	1.93	1.96	1.96	—
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
10	L_2	7.05×10^{-3}	1.61×10^{-3}	5.90×10^{-4}	1.63×10^{-4}	4.31×10^{-5}	1.13×10^{-5}
		2.12	1.46	1.85	1.92	1.93	—
	L_∞	1.41×10^{-2}	2.57×10^{-3}	1.00×10^{-3}	2.79×10^{-4}	7.34×10^{-5}	1.92×10^{-5}
		2.46	1.36	1.84	1.93	1.93	—

Table 8 represents the pointwise errors and corresponding convergence orders for Example 2 in the L_2 and L_∞ norms for different values of the nonlinear parameter v and progressively refined mesh sizes. It is observed that the errors decrease consistently as the mesh size is refined, and the computed convergence orders approach second-order accuracy in both norms. This confirms that the proposed scheme maintains its expected spatial accuracy while simultaneously capturing the temporal behavior under mesh refinement. Moreover, the results demonstrate that the convergence rates remain stable across a wide range of nonlinearities $v = 1, 2, \ldots, 10$. Minor variations in the convergence order for coarse meshes are observed, particularly for large values of v, but these effects diminish as the mesh is refined. This behavior is consistent with the pre-asymptotic regime and does not affect the overall convergence trend. Table 9 reports pointwise errors for fixed discretization parameters $N_x = 160$ and $M_t = 256$, for various values of α and v. The results indicate that the numerical errors remain uniformly small across all tested values of α and v, demonstrating the stability and robustness of the scheme. As α increases, a slight improvement in accuracy is observed, which is consistent with the smoother behavior of solutions for larger fractional orders. In line with Remark 2, the numerical results in both tables confirm that the convergence behavior and stability of the scheme are essentially independent of the nonlinear parameter v. The comparable error magnitudes and consistent convergence rates across all tested values of v demonstrate that the proposed method effectively handles nonlinear effects without loss of accuracy. Overall, Tables 8 and 9 further validate the theoretical findings, showing that the scheme is second-order accurate, stable, and robust for a wide range of fractional orders and nonlinearities.

Table 9: Pointwise errors for Example 2 in L_2 and L_∞ norms for $N_x = 160, M_t = 256, d = 1$, and different values of α and v.

v	Norms	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$	$\alpha = 0.9$	$\alpha = 1$
1	L_2	1.92×10^{-5}	1.45×10^{-5}	1.24×10^{-5}	1.21×10^{-5}	3.31×10^{-6}
	L_∞	2.72×10^{-5}	2.05×10^{-5}	2.01×10^{-5}	1.98×10^{-5}	5.59×10^{-6}
2	L_2	6.72×10^{-5}	6.49×10^{-5}	6.16×10^{-5}	5.72×10^{-5}	4.78×10^{-5}
	L_∞	1.12×10^{-5}	1.09×10^{-5}	1.04×10^{-5}	9.79×10^{-5}	8.38×10^{-5}
3	L_2	1.92×10^{-5}	1.45×10^{-5}	1.33×10^{-5}	1.31×10^{-5}	1.06×10^{-5}
	L_∞	2.72×10^{-5}	2.05×10^{-5}	1.88×10^{-5}	1.85×10^{-5}	1.60×10^{-5}
4	L_2	5.56×10^{-5}	5.25×10^{-5}	4.74×10^{-5}	4.20×10^{-5}	3.23×10^{-5}
	L_∞	9.23×10^{-5}	8.81×10^{-5}	3.25×10^{-5}	7.27×10^{-5}	5.70×10^{-5}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
10	L_2	4.54×10^{-5}	4.03×10^{-5}	3.25×10^{-5}	2.68×10^{-5}	1.91×10^{-5}
	L_∞	7.65×10^{-5}	6.97×10^{-5}	5.89×10^{-5}	4.98×10^{-5}	3.61×10^{-5}

A numerical comparison of the suggested method with the current numerical approaches documented in [22, 35, 25, 44, 21] is presented in Tables 10–14. The numerical results demonstrate that the proposed method achieves higher accuracy compared with the previously reported schemes. Specifically, the findings support its efficacy and possible benefits for real-world uses across a broad range of v values. The tension parameters of the nonpolynomial spline offer a vital benefit over a conventional cubic B-spline because they can prevent unwarranted spurious oscillations. The key feature of tension spline is the tension parameter μ, which allows local control of the spline shape and can effectively suppress oscillations that often arise in convection-dominated regimes. Tables 6, 10 and 14 demonstrate that our algorithm can be better used than common B-spline algorithms [21, 22].

Table 10: Example 2: L_∞ and L_2 errors for $M_t = 4000$, $\alpha = 0.5$, and $v = d = 1$.

N_x	Method [22]		Method [35]		Present method	
	L_2	L_∞	L_2	L_∞	L_2	L_∞
10			2.26×10^{-2}	3.04×10^{-2}	7.14×10^{-3}	1.01×10^{-2}
20			5.44×10^{-3}	7.70×10^{-3}	4.77×10^{-4}	7.48×10^{-4}
40	1.22×10^{-3}	1.73×10^{-3}	1.22×10^{-3}	1.72×10^{-3}	8.20×10^{-5}	1.31×10^{-4}
80	1.77×10^{-4}	2.53×10^{-4}	1.68×10^{-4}	2.38×10^{-4}	2.09×10^{-5}	3.02×10^{-5}
100	5.22×10^{-5}	7.65×10^{-5}	4.23×10^{-5}	5.99×10^{-5}	1.35×10^{-5}	1.90×10^{-5}

Table 11: Comparison for Example 2 in L_2 and L_∞ norms for different values of α with $M_t = 1000$ and $v = d = 1$.

α	Norms	$N_x = 8$		$N_x = 16$		$N_x = 32$	
		Method [25]	Present	Method [25]	Present	Method [25]	Present
0.5	L_2	3.32×10^{-2}	1.80×10^{-2}	8.14×10^{-3}	1.06×10^{-3}	2.02×10^{-3}	1.31×10^{-4}
	L_∞	4.70×10^{-2}	2.54×10^{-2}	1.15×10^{-2}	1.44×10^{-3}	2.86×10^{-3}	2.26×10^{-4}
0.9	L_2	3.29×10^{-2}	1.78×10^{-2}	8.06×10^{-3}	1.05×10^{-3}	2.00×10^{-3}	1.29×10^{-4}
	L_∞	4.65×10^{-2}	2.52×10^{-2}	1.14×10^{-2}	1.43×10^{-3}	2.83×10^{-3}	2.24×10^{-4}

Table 12: Comparison for Example 2 in L_2 and L_∞ norms for different values of α with $M_t = N_x$ and $v = d = 1$.

α	Norms	$N_x = 10$		$N_x = 20$		$N_x = 40$	
		Method [25]	Present	Method [25]	Present	Method [25]	Present
0.5	L_2	2.11×10^{-2}	1.31×10^{-2}	5.21×10^{-3}	3.25×10^{-3}	1.30×10^{-3}	7.91×10^{-4}
	L_∞	2.88×10^{-2}	1.77×10^{-2}	7.37×10^{-3}	4.37×10^{-3}	1.88×10^{-3}	1.11×10^{-3}
0.9	L_2	2.08×10^{-2}	1.05×10^{-2}	5.15×10^{-3}	2.26×10^{-3}	1.28×10^{-3}	4.55×10^{-4}
	L_∞	2.84×10^{-2}	1.42×10^{-2}	7.28×10^{-3}	3.05×10^{-3}	1.81×10^{-3}	6.43×10^{-4}

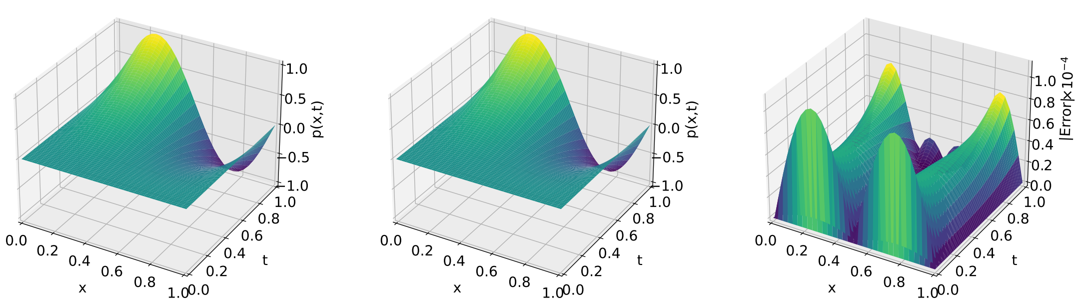
In Figure 9, we display the surface plots of the numerical solution and the absolute error for Example 1 at $\alpha = 0.4, d = v = 1$. All errors in Figures 10–13 are computed using the double mesh principle, based on the difference between the numerical solutions obtained on the meshes $(N_x, M_t) = (64, 128)$ and $(N_x, M_t) = (128, 256)$ for Example 2. In Figures 10–13 illustrate the distributions of absolute errors for different values of α and v.

Table 13: Comparison for Example 2 in L_2 and L_∞ norms for different values of α with $h = 0.01$ and $v = 1, d = 1$.

t	Norms	$\alpha = 0.5$		$\alpha = 1$	
		Method [44]	Present	Method [44]	Present
0.02	L_2	3.764×10^{-3}	1.206×10^{-4}	4.333×10^{-3}	5.332×10^{-5}
	L_∞	8.596×10^{-3}	1.705×10^{-4}	9.738×10^{-3}	7.597×10^{-5}
0.04	L_2	2.577×10^{-3}	5.011×10^{-4}	2.968×10^{-3}	2.123×10^{-4}
	L_∞	5.303×10^{-3}	7.087×10^{-4}	6.010×10^{-3}	3.003×10^{-4}
0.06	L_2	2.166×10^{-3}	1.249×10^{-3}	2.494×10^{-3}	5.998×10^{-4}
	L_∞	5.303×10^{-3}	1.766×10^{-3}	6.010×10^{-3}	8.483×10^{-4}
0.08	L_2	3.921×10^{-3}	2.246×10^{-3}	4.513×10^{-3}	1.223×10^{-3}
	L_∞	8.596×10^{-3}	3.176×10^{-3}	9.738×10^{-3}	1.729×10^{-3}

Table 14: Comparison for Example 2 in L_2 and L_∞ norms for $\alpha = 0.5$ with $N_x = 120, M_t = 2000$, $v = d = 1$ and $T = 0.1$.

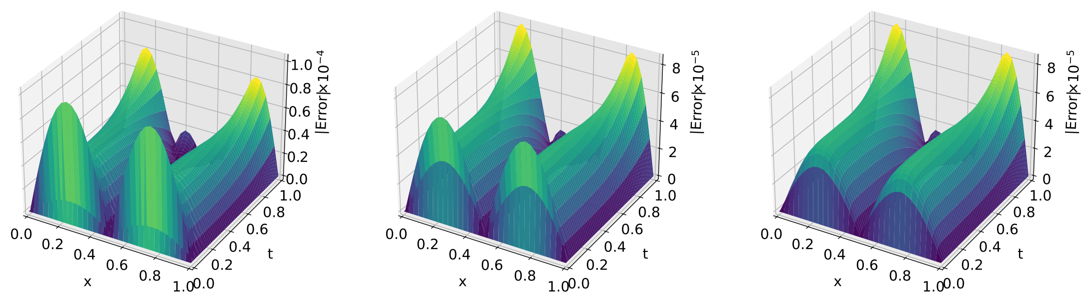
d	Method [21]		Method [25]		Present method	
	L_2	L_∞	L_2	L_∞	L_2	L_∞
0.1	1.501×10^{-5}	2.126×10^{-5}	6.666×10^{-7}	9.427×10^{-7}	1.185×10^{-8}	2.019×10^{-8}
0.01	1.045×10^{-6}	2.001×10^{-6}	1.338×10^{-7}	1.919×10^{-7}	2.744×10^{-8}	4.775×10^{-8}
0.005	7.583×10^{-7}	2.341×10^{-6}	8.573×10^{-8}	1.261×10^{-7}	2.947×10^{-8}	5.143×10^{-8}



(a) $N_x = 64, M_t = 128$ (b) $N_x = 128, M_t = 256$ (c) Error

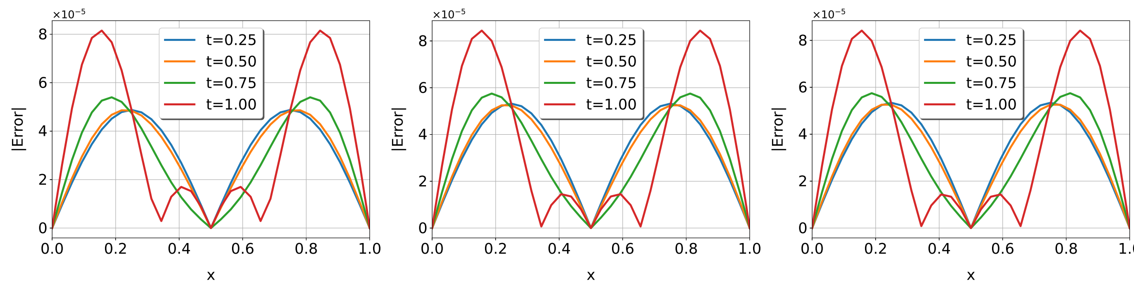
Figure 9: 3D representations of solution behaviors and absolute error corresponding to Example 2 for $\alpha = 0.4$ and $v = d = 1$.

Remark 3. The stability condition, which is $3^\alpha \geq \frac{3}{2}$ and leads to $\alpha \geq 0.37$, is not a necessary condition; rather, it is sufficient. It can be observed from Tables 1 and 2 that for $\alpha = 0.2$, the condition $3^\alpha \geq \frac{3}{2}$ is violated, but the results are of the same precision.



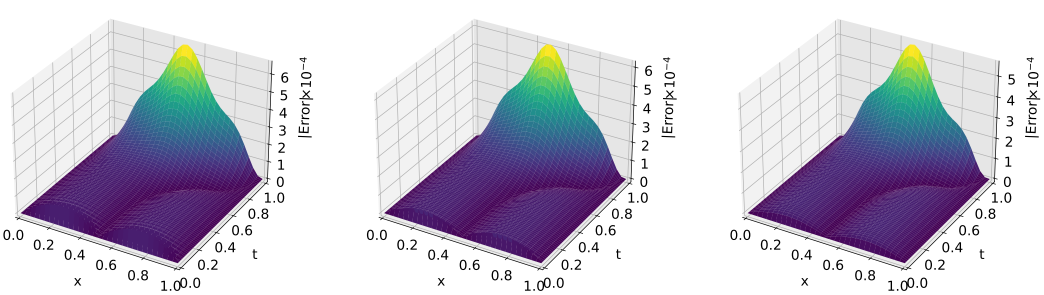
(a) Error for $\alpha = 0.5$ (b) Error for $\alpha = 0.7$ (c) Error for $\alpha = 0.9$

Figure 10: 3D representations of the absolute error behaviors for Example 2 at $v = d = 1$.



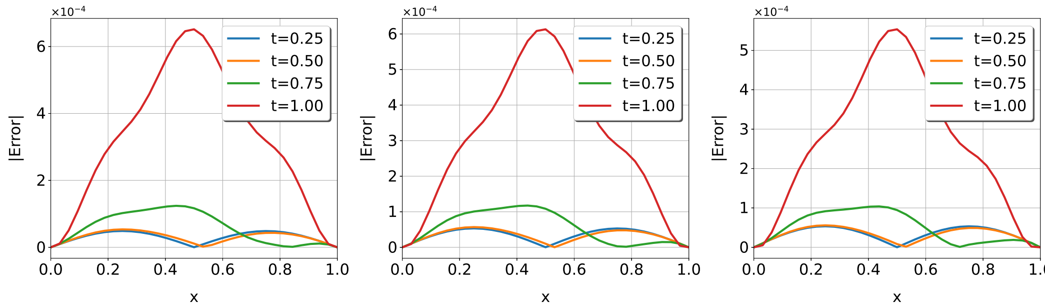
(a) Error for $\alpha = 0.5$ (b) Error for $\alpha = 0.7$ (c) Error for $\alpha = 0.9$

Figure 11: 2D representations of the absolute error behaviors for Example 2 at $v = d = 1$.



(a) Error for $\alpha = 0.5$ (b) Error for $\alpha = 0.7$ (c) Error for $\alpha = 0.9$

Figure 12: 3D representations of the absolute error behaviors for Example 2 at $v = 2, d = 1$.



(a) Error for $\alpha = 0.5$ (b) Error for $\alpha = 0.7$ (c) Error for $\alpha = 0.9$

Figure 13: 2D representations of the absolute error behaviors for Example 2 at $v = 2, d = 1$.

6 Conclusion

This paper develops a numerical method for the generalized nonlinear time-fractional convection-diffusion equation. First, the quasi-linearization process is used to obtain a linear-time fractional convection-diffusion sequence. Next, we apply the scheme on uniform partitions, which consists of the nonpolynomial tension spline scheme in the spatial direction and the Crank–Nicolson discretization in the temporal direction. Analytical and some numerical results from the literature are compared with the results. This comparison demonstrates the effectiveness and efficiency of the proposed approach, which can also be applied to a variety of scientific and physical fields. Additionally, the error norms L_2 and L_∞ are calculated and provided in tables to demonstrate the accuracy of the present approach. The accuracy of the proposed scheme is demonstrated using two test problems for a range of parameter values. The results of these different tests, along with comparisons with other results and the error norms L_2 and L_∞, are displayed in tables and figures. Finally, it has been demonstrated that the present approach is suitable and widely applicable to FDEs. In the future, we will extend this method to Caputo sense time-FDEs near $t = 0$, which exhibit weakly singular behavior.

Acknowledgements

Authors are grateful to there anonymous referees and editor for their constructive comments.

Conflict of interest

The author declares that he has no competing interests.

References

[1] Abolhasani, M., Abbasbandy, S. and Allahviranloo, T. *A new variational iteration method for a class of fractional convection-diffusion equations in large domains*, Math. 5(26), (2017), 1–15.

[2] Ahmad, H., Farooq, M., Khan, I., Nawaz, R., Fewster-Young, N. and Askar, S. *Analysis of nonlinear fractional-order Fisher equation using two reliable techniques*, Open Phys. 22(1), (2024), 1–8.

[3] Akram, T., Abbas, M., Riaz, M.B., Ismail, A.I. and Ali, N.M. *An efficient numerical technique for solving time fractional Burgers equation*, Alex. Eng. J. 59(4), (2020), 2201–2220.

[4] Alam, K.N., Ara, A. and Mahmood, A. *Numerical solutions of time-fractional Burgers equations: A comparison between generalized differential transformation technique and homotopy perturbation method*, Int. J. Numer. Methods Heat Fluid Flow, 22(2), (2012), 175–193.

[5] Atangana, A. *Fractal-fractional differentiation and integration: connecting fractal calculus and fractional calculus to predict complex system*, Chaos Solitons Fractals. 102 (2017), 396–406.

[6] Bekela, A.S., Belachew, M.T. and Wole, G.A. *A numerical method using Laplace-like transform and variational theory for solving time-fractional nonlinear partial differential equations with proportional delay*, Adv. differ. equ. 2020(1), (2020), 1–19.

[7] Bekela, A.S. and Deresse, A.T. *A hybrid yang transform Adomian decomposition method for solving time-fractional nonlinear partial differential equation*, BMC Res. Notes, 17(226), (2024), 1–20.

[8] Bekela, A.S. and Deresse, A.T. *An efficient numerical method for nonlinear time-fractional hyperbolic partial differential equations based on fractional Shehu transform iterative method*, J. Appl. Math. 2025 (2025), 1–22.

[9] Bekela, A.S., Enyadene, L.G. and Woldaregay, M. M. *Analysis and application of a hybrid tension-spline method for nonlinear time-fractional reaction–diffusion equations*, Discov. Appl. Sci. 8(457), (2026), 1–32.

[10] Bekela, A.S. and Woldaregay, M.M. *Formable transform Adomian decomposition method for solving nonlinear time-fractional diffusion equation*, Partial Differ. Equ. Appl. Math. 15 (2025), 1–10.

[11] Bellman, R.E. and Kalaba, R.E. *Quasilinearization and nonlinear boundary-value problems*, Amer. Elsevier Publ. Co., New York, 3, 1965.

[12] Chawla, R., Deswal, K., Kumar, D. and Baleanu, D. *Numerical simulation for generalized time-fractional Burgers' equation with three distinct linearization schemes*, J. Comput. Nonlinear Dyn. 18(4), (2023), 1–16.

[13] Chawla, R., Kumar, D. and Singh, S. *A second-order scheme for the generalized time-fractional Burgers' equation*, J. Comput. Nonlinear Dyn. 19(1), (2024), 1–13.

[14] Choudhary, R., Singh, S. and Kumar, D. *A second-order numerical scheme for the time-fractional partial differential equations with a time delay*, Comput. Appl. Math. 41(3), (2022), 1–28.

[15] Delkhosh, M. *Introduction of derivatives and integrals of fractional order and its applications*, Appl. Math. Phys. 1(4), (2013), 103–119.

[16] Deresse, A.T. and Bekela, A.S. *A deep learning approach: physics-informed neural networks for solving a nonlinear telegraph equation with different boundary conditions*, BMC Res. Notes, 18(77), (2025), 1–21.

[17] Deresse, A.T., Bekela, A.S. and Dufera, T.T. *MT-PINNs: multi-term physics-informed neural networks for solving initial boundary value problems of 2D and 3D nonlinear telegraph equations*, Bound. Value Probl. 2025(146), (2025), 1–29.

[18] Deresse, A.T. and Bekela, A.S. and Dufera, T.T. *Leveraging advanced deep neural networks algorithm for solving multi-dimensional forward and inverse problems of Klein-Gordon equation with quadratic, cubic, and fifth-degree polynomials nonlinearity*, Comput. Math. Model. 36(3), (2025), 489–516.

[19] Du, M., Wang, Z. and Hu, H. *Measuring memory with the order of fractional derivative*, Sci. Rep. 3(1), (2013), 1–3.

[20] Esen, A., Bulut, F. and Oruç, O. *A unified approach for the numerical solution of time-fractional Burgers' type equations*, Eur. Phys. J. Plus, 131(4), (2016), 1–13.

[21] Esen, A. and Tasbozan, O. *Numerical solution of time-fractional Burgers equation*, Acta Univ. Sapientiae Math. 7(2), (2015), 167–185.

[22] Esen, A. and Tasbozan, O. *Numerical solution of time-fractional Burgers equation by cubic B-spline finite elements*, Mediterr. J. Math. 13(3), (2016), 1325–1337.

[23] Fang, J., Nadeem, M., Habib, M., Karim, S. and Wahash, H.A. *A new iterative method for the approximate solution of Klein-Gordon and Sine-Gordon equations*, J. Funct. Spaces, 2022(2), (2022), 1–9.

[24] Gowrisankar, S. and Natesan, S. *An efficient robust numerical method for singularly perturbed Burgers' equation*, Appl. Math. Comput. 346(1), (2019), 385–394.

[25] Guan, J., Rahman, K. and Guo, Z. *Radial basis function finite difference method for solving the generalized time-fractional Burgers equation with three types of boundary conditions*, Chaos Solitons Fractals. 199(3), (2025), 116754.

[26] Hashmi, M.S., Wajiha, M., Yao, S., Ghaffar, A. and Inc, M. *Cubic spline based differential quadrature method: A numerical approach for fractional Burger equation*, Results Phys. 26 (2021), 1–11.

[27] Inc, M. *The approximate and exact solutions of the space-and time-fractional Burgers equations with initial conditions by variational iteration method*, J. Math. Anal. Appl. 345(1), (2008), 476–484.

[28] Jacobs, B.A. and Harley, C. *Application of nonlinear time-fractional partial differential equations to image processing via hybrid Laplace transform method*, J. Math. 2018 (2018), 1–9.

[29] Kanth, A.S.V.R., Aruna, K., Raghavendar, K., Rezazadeh, H. and Inc, M. *Numerical solutions of nonlinear time fractional Klein-Gordon equation via natural transform decomposition method and iterative Shehu transform method*, J. Ocean Eng. Sci. (2021).

[30] Kanth, A.S.V.R. and Sirswal, D. *Analysis and numerical simulation for a class of time-fractional diffusion equation via tension spline*, Numer. Algor. 79(2), (2018), 479–497.

[31] Kumar, R. and Jain, S. *Time fractional generalized Korteweg-de Vries equation: Explicit series solutions and exact solutions*, J. Frac. Calc. Nonlinear Sys. 1(1), (2021), 62–77.

[32] Li, D., Zhang, C. and Ran, M. *A linear finite difference scheme for generalized time fractional Burgers equation*, Appl. Math. Model. 40 (11-12), (2016), 6069–6081.

[33] Majeed, A., Kamran, M. and Rafique, M. *An approximation to the solution of time-fractional modified Burgers' equation using extended cubic B-spline method*, Comput. Appl. Math. 39(4), (2020), 1–21.

[34] Naeem, M., Aljahdaly, N.H., Shah, R. and Weera, W. *The study of fractional-order convection–reaction-diffusion equation via an Elzake Atangana–Baleanu operator*, AIMS Math. 7(10), (2022), 18080–18098.

[35] Onal, M. and Esen, A. *A Crank–Nicolson approximation for the time-fractional Burgers equation*, Appl. math. nonlinear sci. 5(2), (2020), 177–184.

[36] Oru, O., Esen, A. and Bulut, F. *A unified finite difference Chebyshev wavelet method for numerically solving time-fractional Burgers' equation*, Discrete Contin. Dyn. Syst., Ser. S. 12(3), (2019), 533–542.

[37] Podlubny, I. *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, Elsevier, 1998.

[38] Poojitha, S. and Awasthi, A. *Eloquent numerical approach for solving generalized time-fractional convection-diffusion-reaction problems*, Phys. Scr. 99(12), (2024), 1–19.

[39] Priyendhu, K.S., Prakash, P. and Lakshmanan, M. *Invariant subspace method to the initial and boundary value problem of the higher dimensional nonlinear time-fractional PDEs*, Commun. Nonlinear Sci. Numer. Simul. 122 (2023), 107245.

[40] Rubin, S.G. and Graves, J.R.A. *Viscous flow solutions with a cubic spline approximation*, Comput. Fluids, 3(1), (1975), 1–36.

[41] Salama, F.M. *An efficient explicit group method for time-fractional Burgers equation*, Front. Phys. 13, (2025), 1–18.

[42] Sugimoto, N. *Burgers equation with a fractional derivative; hereditary effects on nonlinear acoustic waves*, J. Fluid Mech. 225 (1991), 631–653.

[43] Vijayaram, S. and Balasubramaniam, P. *Optimal control for nonlinear time-fractional Schrudinger equation: an application to quantum optics*, Phys. Scr. 99(9), (2024), 095115.

[44] Yousif, M.A. and Hamasalh, F.K. *Novel simulation of the time-fractional Burgers-Fisher equations using a nonpolynomial spline fractional continuity method*, AIP Adv. 12(11), (2022), 1–13.