frontiers in Physics

ORIGINAL RESEARCH
published: 23 September 2020
doi: 10.3389/fphy.2020.00288

Numerical Treatment of Time-Fractional Klein–Gordon Equation Using Redefined Extended Cubic B-Spline Functions

Muhammad Amin[1,2], *Muhammad Abbas*[3,4*], *Muhammad Kashif Iqbal*[5] and *Dumitru Baleanu*[6,7,8]

[1] Department of Mathematics, National College of Business Administration & Economics, Lahore, Pakistan, [2] Department of Mathematics, University of Sargodha, Sargodha, Pakistan, [3] Informetrics Research Group, Ton Duc Thang University, Ho Chi Minh City, Vietnam, [4] Faculty of Mathematics and Statistics, Ton Duc Thang University, Ho Chi Minh City, Vietnam, [5] Department of Mathematics, Government College University, Faisalabad, Pakistan, [6] Department of Mathematics, Faculty of Arts and Sciences, Cankaya University, Ankara, Turkey, [7] Department of Medical Research, China Medical University, Taichung, Taiwan, [8] Institute of Space Sciences, Bucharest, Romania

In this article we develop a numerical algorithm based on redefined extended cubic B-spline functions to explore the approximate solution of the time-fractional Klein–Gordon equation. The proposed technique employs the finite difference formulation to discretize the Caputo fractional time derivative of order $\alpha \in (1, 2]$ and uses redefined extended cubic B-spline functions to interpolate the solution curve over a spatial grid. A stability analysis of the scheme is conducted, which confirms that the errors do not amplify during execution of the numerical procedure. The derivation of a uniform convergence result reveals that the scheme is $O(h^2 + \Delta t^{2-\alpha})$ accurate. Some computational experiments are carried out to verify the theoretical results. Numerical simulations comparing the proposed method with existing techniques demonstrate that our scheme yields superior outcomes.

Keywords: redefined extended cubic B-spline, time fractional Klein-Gorden equation, Caputo fractional derivative, finite difference method, convergence analysis

OPEN ACCESS

Edited by:
Jordan Yankov Hristov, University of Chemical Technology and Metallurgy, Bulgaria

Reviewed by:
Praveen Agarwal, Anand International College of Engineering, India
Hossein Jaferi, University of South Africa, South Africa

***Correspondence:**
Muhammad Abbas
muhammadabbas@tdtu.edu.vn

Specialty section:
This article was submitted to Mathematical and Statistical Physics, a section of the journal Frontiers in Physics

Received: *02 March 2020*
Accepted: *25 August 2020*
Published: *23 September 2020*

Citation:
Amin M, Abbas M, Iqbal MK and Baleanu D (2020) Numerical Treatment of Time-Fractional Klein–Gordon Equation Using Redefined Extended Cubic B-Spline Functions. Front. Phys. 8:288. doi: 10.3389/fphy.2020.00288

1. INTRODUCTION

The subject of fractional-order differential equations has attracted considerable interest due to its applications in a wide range of fields, such as traffic flow, earthquakes and other physical phenomena, signal processing, finance, control theory, fractional dynamics, and mathematical modeling [1–10]. In recent years, the analytical and numerical study of fractional-order differential equations has become a dynamic area of research. Several numerical and analytical techniques have been developed to handle these types of equations [11–22]. There are a number of different definitions of fractional-order derivatives, with different applications. An excellent overview can be found in the works [23–31]. This article is concerned with the following time-fractional non-linear Klein–Gordon equation (KGE):

$$\frac{\partial^\alpha}{\partial t^\alpha} v(x,t) + \rho \frac{\partial^2}{\partial x^2} v(x,t) + \rho_1 v(x,t) + \rho_2 v^\sigma(x,t) = f(x,t), \qquad 0 < x \leq L,\ t_0 < t \leq T, \tag{1}$$

$$v(x, t_0) = \varphi_1(x), \quad v_t(x, t_0) = \varphi_2(x), \tag{2}$$

$$v(0, t) = \varphi_3(t), \quad v(L, t) = \varphi_4(t), \tag{3}$$

where $\frac{\partial^\alpha}{\partial t^\alpha}$ represents the Caputo fractional time derivative, $v = v(x, t)$ denotes the displacement of the wave at (x, t), $\alpha \in (1, 2]$ is the fractional order of the time derivative, $f(x, t)$ is the source term, ρ, ρ_1 and ρ_2 are real numbers, and $\sigma = 2$ or 3.

The fractional KGE plays a significant role in quantum mechanics, the study of solitons, and condensed matter physics. Many approaches have been adopted to solve equations of Klein/sine–Gordon type efficiently, including the Adomian decomposition method, the variational iteration method [32–34], and the homotopy analysis method [35]; see also the references cited in these works. Jafari et al. proposed using fractional B-splines for approximate solution of fractional differential equations [36]. In Vong and Wang [37, 38] space compact difference schemes were applied to one- and two-dimensional time-fractional Klein–Gordon-type equations, and stability and convergence of the proposed numerical approaches were established with the aid of an energy method. In Dehghan et al. [39] the authors used a meshless method based on radial basis functions to develop an unconditionally stable numerical scheme for fractional Klein/sine–Gordon equations. The Adomian decomposition method and an iterative method were applied in Jafari [40] to solve Klein–Gordon-type equations involving fractional time derivatives. A fully spectral approach was employed in Chen et al. [41] that uses finite differences for time discretization and Legendre spectral approximation in the spatial direction to construct numerical solutions of non-linear partial differential equations involving fractional derivatives. A sinc–Chebyshev collocation method (SCCM) was developed in Nagy [42] for numerical treatment of the time-fractional non-linear KGE. Recently, in Kanwal et al. [43], Genocchi polynomials were employed together with the Ritz–Galerkin scheme to solve fractional KGEs and diffusion wave equations. A linearized second-order scheme was introduced in Lyu and Vong [44] to solve non-linear time-fractional Klein–Gordon-type equations. Later on, in Doha et al. [45], a space–time spectral approximation was proposed for solving non-linear variable-order fractional Klein/sine–Gordon differential equations.

In this article we propose using redefined extended cubic B-spline (RECBS) functions for numerical solution of the time-fractional KGE. RECBS functions are basically a generalization of typical cubic B-spline functions that involve a free parameter which provides the flexibility to fine-tune the solution curve. We employ the usual finite central difference approach to discretize the Caputo fractional time derivative and use RECBS functions for spatial integration.

This article is organized as follows. The Caputo definition of fractional time derivative and the finite difference formulation for temporal discretization are reviewed in section 2; this section also includes a brief introduction to extended cubic B-spline and RECBS functions and their applications to space discretization. The stability analysis of the proposed algorithm is presented in section 3, and the description of theoretical convergence is given in section 4. The approximate results are reported and discussed in section 5. Finally, concluding remarks are given in section 6.

2. DESCRIPTION OF NUMERICAL TECHNIQUE

2.1. Time Discretization

Let the time domain $[0, T]$ be divided into R subintervals of equal length $\Delta t = \frac{T}{R}$ with endpoints $0 = t_0 < t_1 < \cdots < t_R = T$, where $t_r = r\Delta t$ and $r = 0:1:R$. We first discretize the Caputo fractional derivative at $t = t_{r+1}$ as [46]

$$\begin{aligned}
\frac{\partial^\alpha v(x, t_{r+1})}{\partial t^\alpha} &= \frac{1}{\Gamma(2-\alpha)} \int_0^{t_k} \frac{\partial^2 v(x, w)}{\partial w^2} (t_{r+1} - w)^{-\alpha+1}\, dw \quad (1 < \alpha \le 2) \\
&= \frac{1}{\Gamma(2-\alpha)} \sum_{k=0}^{r} \int_{t_k}^{t_{k+1}} \frac{\partial^2 v(x, w)}{\partial w^2} (t_{r+1} - w)^{-\alpha+1}\, dw. \\
&= \frac{1}{\Gamma(2-\alpha)} \sum_{k=0}^{r} \frac{v(x, t_{k+1}) - 2v(x, t_k) + v(x, t_{k-1})}{\Delta t^2} \int_{t_k}^{t_{k+1}} (t_{r+1} - w)^{-\alpha+1}\, dw + l_{\Delta t}^{r+1} \\
&= \frac{1}{\Gamma(2-\alpha)} \sum_{k=0}^{r} \frac{v(x, t_{k+1}) - 2v(x, t_k) + v(x, t_{k-1})}{\Delta t^2} \int_{t_{r-k}}^{t_{r-k+1}} (\epsilon)^{-\alpha+1}\, d\epsilon + l_{\Delta t}^{r+1} \\
&= \frac{1}{\Gamma(2-\alpha)} \sum_{k=0}^{r} \frac{v(x, t_{r-k+1}) - 2v(x, t_{r-k}) + v(x, t_{r-k-1})}{\Delta t^2} \int_{t_k}^{t_{k+1}} (\epsilon)^{-\alpha+1}\, d\epsilon + l_{\Delta t}^{r+1} \\
&= \frac{1}{\Gamma(3-\alpha)} \sum_{k=0}^{r} \frac{v(x, t_{r-k+1}) - 2v(x, t_{r-k}) + v(x, t_{r-k-1})}{\Delta t^\alpha} ((k+1)^{2-\alpha} - k^{2-\alpha}) + l_{\Delta t}^{r+1} \\
&= \frac{1}{\Gamma(3-\alpha)} \sum_{k=0}^{r} p_k \frac{v(x, t_{r-k+1}) - 2v(x, t_{r-k}) + v(x, t_{r-k-1})}{\Delta t^\alpha} + l_{\Delta t}^{r+1},
\end{aligned} \tag{4}$$

where $p_k = (k+1)^{2-\alpha} - k^{2-\alpha}$, $\epsilon = (t_{r+1} - w)$, and $l_{\Delta t}^{r+1}$ is the truncation error. The truncation error is bounded, i.e.,

$$|l_{\Delta t}^{r+1}| \le \psi(\Delta t)^{2-\alpha}, \tag{5}$$

where ψ is a constant. The coefficients p_k in (4) possess the following attributes:

- the p_k's are non-negative for $k = 0, 1, 2, \ldots, r$;
- $1 = p_0 > p_1 > p_2 > p_3 > \cdots > p_n$, and $p_n \to 0$ as $n \to \infty$;
- $(2p_0 - p_1) + \sum_{k=1}^{r-1}(-p_{k+1} + 2p_k - p_{k-1}) + (2p_r - p_{r-1}) - p_r = 1$.

Substituting Equation (4) into Equation (1), we get

$$\frac{1}{\Gamma(3-\alpha)(\Delta t)^{\alpha}} \sum_{k=0}^{r} p_k \big[v(x, t_{r-k+1}) - 2v(x, t_{r-k}) + v(x, t_{r-k-1})\big] + \rho v_{xx}(x,t) + \rho_1 v(x,t) + \rho_2 v^{\sigma}(x,t) = f(x,t) \quad (r = 0, 1, 2, \ldots, R-1). \tag{6}$$

Suppose $\beta = \frac{1}{\Gamma(3-\alpha)(\Delta t)^{\alpha}}$ and $v(x, t_{r+1}) = v^{r+1}$. Applying a θ-weighted scheme, Equation (6) takes the form

$$\beta p_0(v^{r+1} - 2v^r + v^{r-1}) + \beta \sum_{k=1}^{r} p_k(v^{r-k+1} - 2v^{r-k} + v^{r-k-1}) + \theta(\rho v_{xx}^{r+1} + \rho_1 v^{r+1}) = f^{r+1} - (1-\theta)(\rho v_{xx}^{r} + \rho_1 v^r) - \rho_2 (v^{\sigma})^r \quad (r = 0, 1, 2, \ldots, R-1). \tag{7}$$

For $\theta = 1$, we obtain the following semi-discretized numerical scheme:

$$(\beta p_0 + \rho_1) v^{r+1} + \rho v_{xx}^{r+1} = 2\beta p_0 v^r + \beta \sum_{k=1}^{r} p_k (v^{r-k+1} - 2v^{r-k} + v^{r-k-1}) - \rho_2 (v^{\sigma})^r - \beta p_0 v^{r-1} + f^{r+1} (r = 0, 1, 2, \ldots, R-1). \tag{8}$$

2.2. Extended Cubic B-Spline Functions

Let the spatial domain $[a, b]$ be partitioned into M parts of equal length $h = \frac{b-a}{M}$ with boundary points $a = x_0 < x_1 < \cdots < x_M = b$, where $x_m = x_0 + mh$ for $m = 0:1:M$. For a sufficiently continuous function $v(x,t)$, there always exists a unique extended cubic B-spline (ECBS) approximation $V^*(x,t)$:

$$V^*(x,t) = \sum_{m=-1}^{M+1} \xi_m(t) S_m(x, \lambda), \tag{9}$$

where the $\xi_m(t)$ are to be calculated and the fourth-degree ECBS blending functions $S_m(x,\lambda)$ are defined as [47]

$$S_m(x,\lambda) = \frac{1}{24h^4} \begin{cases} 4h(x - x_{m-2})^3(1-\lambda) + 3(x - x_{m-2})^4 \lambda & \text{if } x \in [x_{m-2}, x_{m-1}), \\ h^4(4-\lambda) + 12h^3(x - x_{m-1}) + 6h^2(x - x_{m-1})^2(2+\lambda) \\ \quad - 12h(x - x_{m-1})^3 - 3(x - x_{m-1})^4 \lambda & \text{if } x \in [x_{m-1}, x_m), \\ h^4(4-\lambda) - 12h^3(x - x_{m+1}) - 6h^2(x - x_{m+1})^2(2+\lambda) \\ \quad + 12h(x - x_{m+1})^3 + 3(x - x_{m-1})^4 \lambda & \text{if } x \in [x_m, x_{m+1}), \\ -4h(x - x_{m+2})^3(1-\lambda) - 3(x - x_{m+2})^4 \lambda & \text{if } x \in [x_{m+1}, x_{m+2}), \\ 0 & \text{otherwise}. \end{cases} \tag{10}$$

Here λ, with $-n(n-2) \leq \lambda \leq 1$, is a real number responsible for fine-tuning the curve, and n gives the degree of the ECBS used to generate different forms of ECBS functions. The approximate solution $(V^*)_m^r = V^*(x_m, t^r)$ and its first two derivatives with respect to the spatial variable x at the rth time step can be expressed in terms of ξ_m as [48]

$$\begin{cases} (V^*)_m^r = b_1 \xi_{m-1}^r + b_2 \xi_m^r + b_1 \xi_{m+1}^r, \\ (V_x^*)_m^r = b_3 \xi_{m-1}^r - b_3 \xi_{m+1}^r, \\ (V_{xx}^*)_m^r = b_4 \xi_{m-1}^r + b_5 \xi_m^r + b_4 \xi_{m+1}^r, \end{cases} \tag{11}$$

where $b_1 = \frac{4-\lambda}{24}$, $b_2 = \frac{16+2\lambda}{24}$, $b_3 = \frac{-1}{2h}$, $b_4 = \frac{2+\lambda}{2h^2}$, and $b_5 = \frac{-4-2\lambda}{2h^2}$.

2.3. Redefined Extended Cubic B-Spline Functions

In the typical ECBS collocation method, the basis functions $S_{-1}, S_0, \ldots, S_{M+1}$ do not vanish at the boundaries of the spatial domain when Dirichlet-type end conditions are imposed. Therefore, we need to redefine them so that the resulting set of basis functions will vanish at the boundaries. For this, a weight function $\Phi(x,t)$ is introduced to eliminate ξ_{-1} and ξ_{M+1} from Equation (9) in the following manner [49]:

$$V(x,t) = \Phi(x,t) + \sum_{m=0}^{M} \xi_m(t) \tilde{S}_m(x,\lambda), \tag{12}$$

where the weight function $\Phi(x,t)$ and the redefined ECBS (RECBS) functions are given by

$$\Phi(x,t) = \frac{S_{-1}(x,\lambda)}{S_{-1}(x_0,\lambda)} \varphi_3(t) + \frac{S_{M+1}(x,\lambda)}{S_{M+1}(x_M,\lambda)} \varphi_4(t) \tag{13}$$

and.

$$\begin{cases} \tilde{S}_m(x,\lambda) = S_m(x,\lambda) - \frac{S_m(x_0,\lambda)}{S_{-1}(x_0,\lambda)} S_{-1}(x,\lambda) & \text{for } m = 0, 1, \\ \tilde{S}_m(x,\lambda) = S_m(x,\lambda) & \text{for } m = 2:1:M-2, \\ \tilde{S}_m(x,\lambda) = S_m(x,\lambda) - \frac{S_m(x_M,\lambda)}{S_{M+1}(x_M,\lambda)} S_{M+1}(x,\lambda) & \text{for } m = M-1, M. \end{cases} \tag{14}$$

2.4. Space Discretization

Using Equation (12) in Equation (8) at $t = t_{r+1}$, we obtain

$$(\beta p_0 + \rho_1) V^{r+1} + \rho V_{xx}^{r+1} = 2\beta p_0 V^r + \beta \sum_{k=1}^{r} p_k (V^{r-k+1} - 2V^{r-k} + V^{r-k-1}) - \rho_2 (V^{\sigma})^r - \beta p_0 V^{r-1} + f^{r+1}. \tag{15}$$

Discretizing at $x = x_j$, we get

$$(\beta + \rho_1) V_j^{r+1} + \rho (V_{xx})_j^{r+1} = 2\beta V_j^r + \beta \sum_{k=1}^{r} p_k (V_j^{r-k+1} - 2V_j^{r-k}$$

$$+ V_j^{r-k-1}) - \rho_2(V^{\sigma})_j^r - \beta V_j^{r-1} + f_j^{r+1} (j = 0, 1, 2, \ldots, M). \tag{16}$$

Using (12), the last expression takes the form

$$\begin{aligned}(\beta + \rho_1)\left[\Phi_j^{r+1} + \sum_{m=0}^{M} \xi_m^{r+1} \tilde{S}_m(x_j, \lambda)\right] + \rho\Big[(\Phi_{xx})_j^{r+1} \\ + \sum_{m=0}^{M} \xi_m^{r+1} \tilde{S}_m(x_j, \lambda)\Big] \\ = 2\beta V_j^r + \beta \sum_{k=1}^{r} p_k (V_j^{r-k+1} - 2V_j^{r-k} + V_j^{r-k-1}) \\ - \rho_2(V^{\sigma})_j^r - \beta V_j^{r-1} + f_j^{r+1} \\ (j = 0, 1, 2, \ldots, M).\end{aligned} \tag{17}$$

Consequently, we get the following system of $M + 1$ equations in $M + 1$ unknowns:

$$\begin{pmatrix} a_1^* & & & & & & \\ a_1 & a_2 & a_1 & & & & \\ & a_1 & a_2 & a_1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & a_1 & a_2 & a_1 & \\ & & & & a_1 & a_2 & a_1 \\ & & & & & & a_1^* \end{pmatrix} \begin{pmatrix} \xi_0^{r+1} \\ \xi_1^{r+1} \\ \vdots \\ \vdots \\ \xi_{M-1}^{r+1} \\ \xi_M^{r+1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ y_{M-1} \\ y_M \end{pmatrix}, \tag{18}$$

where

$$\begin{aligned} a_1^* &= \frac{12\rho(\lambda + 2)}{h^2(\lambda - 4)}, \quad a_1 = \frac{h^2(\beta + \rho_1)(\lambda - 4) + 12\rho(\lambda + 2)}{24h^2}, \\ a_2 &= \frac{h^2(\beta + \rho_1)(\lambda + 8) - 12\rho(\lambda + 2)}{12h^2}, \\ y_j &= 2\beta V_j^r + \beta \sum_{k=1}^{r} p_k (V_j^{r-k+1} - 2V_j^{r-k} + V_j^{r-k-1}) \\ &\quad - \rho_2(V^{\sigma})_j^r - \beta V_j^{r-1} + \Psi_j^{r+1}, \\ \Psi_j^r &= f_j^r - (\beta + \rho_1)\Phi_j^r - \rho(\Phi_{xx})_j^r. \end{aligned}$$

To start the numerical procedure, we use the given initial conditions to obtain the set of equations

$$\begin{cases} (V')_m^0 = \varphi_1'(x_m) & \text{for } m = 0, \\ (V)_m^0 = \varphi_1(x_m) & \text{for } m = 1:1:M-1, \\ (V')_m^0 = \varphi_1'(x_m) & \text{for } m = M. \end{cases} \tag{19}$$

The matrix representation of (19) is

$$\begin{pmatrix} b_1{}^* & b_2{}^* & & & & & \\ b_1 & b_2 & b_1 & & & & \\ & b_1 & b_2 & b_1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & b_1 & b_2 & b_1 & \\ & & & & b_1 & b_2 & b_1 \\ & & & & & -b_2{}^* & -b_1{}^* \end{pmatrix} \begin{pmatrix} \xi_0^0 \\ \xi_1^0 \\ \vdots \\ \vdots \\ \xi_{M-1}^0 \\ \xi_M^0 \end{pmatrix} \tag{20}$$

$$= \begin{pmatrix} (\varphi_1')_0 - (\Phi')_0^0 \\ (\varphi_1)_1 - \Phi_1^0 \\ \vdots \\ \vdots \\ (\varphi_1)_{M-1} - \Phi_{M-1}^0 \\ (\varphi_1')_M - (\Phi')_M^0 \end{pmatrix},$$

where $b_1{}^* = \frac{8+\lambda}{h(4-\lambda)}$ and $b_2{}^* = \frac{1}{h}$. We solve (20) to obtain $[\xi_0^0, \xi_1^0, \ldots, \xi_M^0]^T$. The ξ_j values are then substituted into (12) to get V^0. Now we can use (18) for $r = 0, 1, 2, \ldots, R - 1$. However, for $r = 0$ the term involving V^{-1} appears in Equation (18). This issue is resolved by using the following substitution derived from the velocity condition given in (2):

$$V^{-1} = V^0 - \Delta t \phi_2(x).$$

3. STABILITY ANALYSIS

We use the Fourier method to study the stability of the proposed numerical method. Let ε_m^r and $\tilde{\varepsilon}_m^r$ denote, respectively, the exact and approximate growth factors of the Fourier modes. The error, ϱ_m^r, is given by

$$\varrho_m^r = \varepsilon_m^r - \tilde{\varepsilon}_m^r, \qquad m = 1:1:M-1, \quad r = 0:1:R, \tag{21}$$

where $\varrho^r = [\varepsilon_1^r, \varepsilon_2^r, \ldots, \varepsilon_{M-1}^r]^T$.

For the sake of simplicity, we shall investigate the stability of the proposed scheme with $f = 0$. The equation for the round-off error is derived from Equations (8) and (21) as

$$\begin{aligned} &(\beta b_1 + \rho_1 b_1 + \rho b_4)\varrho_{m-1}^{r+1} + (\beta b_2 + \rho_1 b_2 + \rho b_5)\varrho_m^{r+1} \\ &+ (\beta b_1 + \rho_1 b_1 + \rho b_4)\varrho_{m+1}^{r+1} \\ &= 2\beta\big(b_1\varrho_{m-1}^r + b_2\varrho_m^r + b_1\varrho_{m+1}^r\big) - \beta\big(b_1\varrho_{m-1}^{r-1} + b_2\varrho_m^{r-1} \\ &+ b_1\varrho_{m+1}^{r-1}\big) \\ &\quad - \beta\sum_{k=1}^{r} p_k\Big[b_1\big(\varrho_{m-1}^{r-k+1} - 2\varrho_{m-1}^{r-k} + \varrho_{m-1}^{r-k-1}\big) \\ &+ b_2\big(\varrho_m^{r-k+1} - 2\varrho_m^{r-k} + \varrho_m^{r-k-1}\big) \\ &\qquad + b_1\big(\varrho_{m+1}^{r-k+1} - 2\varrho_{m+1}^{r-k} + \varrho_{m+1}^{r-k-1}\big)\Big]. \end{aligned} \tag{22}$$

The error equation satisfies the end conditions

$$\varrho_m^0 = \varphi_1(x_m), \quad m = 1:1:M, \tag{23}$$

and

$$\varrho_0^r = \varphi_3(t_r), \quad \varrho_M^r = \varphi_4(t_r), \quad r = 0:1:R. \tag{24}$$

We define the grid function as

$$\varrho^r = \begin{cases} \varrho_m^r & \text{if } x_m - \frac{h}{2} < x \le x_m + \frac{h}{2}, \text{ for } m = 1:1:M-1, \\ 0 & \text{if } a \le x \le \frac{2a+h}{2} \text{ or } \frac{2b-h}{2} \le x \le b. \end{cases} \tag{25}$$

Now, $\varrho^r(x)$ can be written in the form of a Fourier series as follows:

$$\varrho^r(x) = \sum_{r=-\infty}^{\infty} \varepsilon_r(n) e^{\frac{2\pi \iota n x}{b-a}}, \quad r = 1:1:R, \tag{26}$$

where

$$\varepsilon_r(n) = \frac{1}{b-a} \int_a^b \varrho^r(x) e^{\frac{-2\pi \iota n x}{b-a}}\, dx. \tag{27}$$

Taking the $\|\cdot\|_2$ norm, we get

$$\begin{aligned}\|\varrho^r\|_2 &= \left(\sum_{n=1}^{R-1} h|\varrho_n^r|^2\right)^{\frac{1}{2}} \\ &= \left(\int_a^{a+\frac{h}{2}} |\varrho^r|^2\, dx + \sum_{n=1}^{R-1} \int_{x_n-\frac{h}{2}}^{x_n+\frac{h}{2}} |\varrho^r|^2\, dx + \int_{b-\frac{h}{2}}^{b} |\varrho^r|^2\, dx\right)^{\frac{1}{2}} \\ &= \left(\int_a^b |\varrho^r|^2\, dx\right)^{\frac{1}{2}}.\end{aligned}$$

From Parseval's equality we have $\int_a^b |\varrho^r(n)|^2\, dx = \sum_{-\infty}^{\infty} |\varepsilon_n(m)|^2$, so the above expression can be written as

$$\|\varrho^r\|_2^2 = \sum_{r=-\infty}^{\infty} |\varepsilon_r(n)|^2. \tag{28}$$

Next, we consider the solution in terms of Fourier series,

$$\varrho_k^r = \varepsilon_r e^{\iota \nu k h}, \tag{29}$$

where $\iota = \sqrt{-1}$ and $\nu = \frac{2\pi n}{b-a}$. Using Equation (29) in Equation (22) and then dividing by $e^{i\nu kh}$ gives

$$\begin{aligned}&(\beta b_1 + \rho_1 b_1 + \rho b_4)\varepsilon_{r+1} e^{-\iota \nu h} + (\beta b_2 + \rho_1 b_2 + \rho b_5)\varepsilon_{r+1} \\ &+ (\beta b_1 + \rho_1 b_1 + \rho b_4)\varepsilon_{r+1} e^{\iota \nu h} \\ &= 2\beta\big(b_1 \varepsilon_r e^{-\iota \nu h} + b_2 \varepsilon_r + b_1 \varepsilon_r e^{i\nu h}\big) - \beta\big(b_1 \varepsilon_{r-1} e^{-\iota \nu h} \\ &+ b_2 \varepsilon_{r-1} + b_1 \varepsilon_{r-1} e^{i\nu h}\big) \\ &\quad - \beta \sum_{k=1}^{r} p_k \Big[b_1\big(\varepsilon_{r-k+1} e^{-\iota \nu h} - 2\varepsilon_{r-k} + \varepsilon_{r-k-1} e^{\iota \nu h}\big) \\ &\qquad + b_2\big(\varepsilon_{r-k+1} - 2\varepsilon_{r-k} + \varepsilon_{r-k-1}\big) \\ &+ b_1\big(\varepsilon_{r-k+1} e^{-\iota \nu h} - 2\varepsilon_{r-k} e^{\iota \nu h} + \varepsilon_{r-k-1} e^{\iota \nu h}\big)\Big].\end{aligned} \tag{30}$$

We know that $e^{i\nu h} + e^{-i\nu h} = 2\cos(\nu h)$, so after collecting like terms, the following useful relation is obtained:

$$\varepsilon_{r+1} = \frac{1}{\eta}\left[2\varepsilon_r - \varepsilon_{r-1} - \sum_{k=1}^{r} p_k\big(\varepsilon_{r-k+1} - 2(b_1+b_2)\varepsilon_{r-k} + \varepsilon_{r-k-1}\big)\right], \tag{31}$$

where $\eta = 1 + \frac{\rho 1}{\beta} + \frac{12\rho(2+\nu)\sin^2(\nu h/2)}{\beta h^2\{-6+(4-\nu)\sin^2(\nu h/2)\}}$. Now it is obvious that $\eta \geq 1$ for $\nu > -2$.

TABLE 2 | Absolute and relative errors for Example 5.1 with $M = 100$, $\Delta t = 0.001$, and $\alpha = 1.6$.

		SCCM [42]		Proposed method	
t	x	L_∞	L_2	L_∞	L_2
	0.4	9.3726×10^{-4}	1.3282×10^{-2}	1.6174×10^{-5}	1.2207×10^{-5}
0.4	0.6	9.4592×10^{-4}	1.6950×10^{-2}	6.3939×10^{-6}	1.1035×10^{-6}
	0.8	6.5448×10^{-4}	1.4462×10^{-1}	5.1612×10^{-6}	3.2573×10^{-6}
	0.4	1.7359×10^{-4}	8.6999×10^{-4}	2.4030×10^{-5}	9.1532×10^{-6}
0.8	0.6	1.2080×10^{-4}	1.6683×10^{-3}	6.7766×10^{-6}	2.8126×10^{-6}
	0.8	2.4657×10^{-4}	1.9263×10^{-2}	3.5003×10^{-6}	9.0128×10^{-7}

TABLE 1 | Absolute errors for Example 5.1 with $M = 100$, $\Delta t = 0.001$, and different values of α.

	SCCM [42]			Proposed method		
x	$\alpha = 1.5$	$\alpha = 1.7$	$\alpha = 1.9$	$\alpha = 1.5$	$\alpha = 1.7$	$\alpha = 1.9$
0.1	8.7105×10^{-4}	4.3675×10^{-4}	5.0452×10^{-4}	1.0827×10^{-6}	4.6777×10^{-6}	9.5482×10^{-6}
0.2	8.7781×10^{-4}	9.8359×10^{-4}	7.5328×10^{-5}	9.2126×10^{-6}	1.1035×10^{-6}	3.6308×10^{-5}
0.3	6.2089×10^{-4}	4.8897×10^{-5}	1.1241×10^{-4}	2.9024×10^{-6}	1.2573×10^{-5}	9.1646×10^{-6}
0.4	5.7015×10^{-4}	7.6534×10^{-4}	1.6772×10^{-4}	3.6966×10^{-6}	8.1441×10^{-6}	7.0990×10^{-6}
0.5	5.1476×10^{-4}	9.3043×10^{-4}	2.5022×10^{-4}	8.3386×10^{-6}	2.5203×10^{-7}	2.3918×10^{-5}
0.6	4.8948×10^{-4}	9.4248×10^{-4}	2.5022×10^{-4}	1.0128×10^{-5}	7.3829×10^{-6}	9.8467×10^{-5}
0.7	5.1671×10^{-4}	7.5585×10^{-5}	2.5022×10^{-4}	8.9851×10^{-6}	7.1672×10^{-6}	7.1855×10^{-6}
0.8	5.3919×10^{-4}	5.2006×10^{-4}	2.5022×10^{-4}	5.3467×10^{-6}	7.2518×10^{-6}	3.2774×10^{-5}
0.9	6.0660×10^{-4}	5.4848×10^{-4}	2.5022×10^{-5}	1.7505×10^{-7}	9.7572×10^{-6}	2.8528×10^{-6}

Lemma 3.1. *Let ε_r be the solution of Equation (31). Then $|\varepsilon_r| \leq |\varepsilon_0|$ for $r = 0:1:R$.*

Proof: For $r = 0$ in (31), we have

$$|\varepsilon_1| = \frac{1}{\eta}|\varepsilon_0| \leq |\varepsilon_0| \quad \text{for } \eta \geq 1.$$

Suppose that the result is true for $r = 1:1:R$. Then, from Equation (31) we get

$$\begin{aligned}|\varepsilon_{r+1}| &\leq \frac{1}{\eta}|\varepsilon_r| - \frac{1}{\eta}\sum_{k=1}^{r} p_k\big(|\varepsilon_{r-k+1}| - 2|\varepsilon_{r-k}| + |\varepsilon_{r-k-1}|\big)\\ &\leq \frac{1}{\eta}|\varepsilon_0| - \frac{1}{\eta}|\varepsilon_0| - \sum_{k=1}^{r} p_k\big(|\varepsilon_0| - |\varepsilon_0|\big)\\ &\leq |\varepsilon_0|.\end{aligned}$$

Theorem 1. *The implic it collocation technique presented in Equation (13) is unconditionally stable.*

Proof: Using Lemma (3.1) and Equation (28), we obtain

$$\|\varrho^r\|_2 \leq |\varrho^0|_2, \quad r = 0:1:R.$$

4. CONVERGENCE OF THE SCHEME

To investigate the convergence of the proposed scheme, we follow the approach in Khalid et al. [50]. Before proceeding, we state the following useful theorems [51, 52].

Theorem 2. *Let $\Pi = \{a = x_0, x_1, \ldots, x_M = b\}$ be a partition of $[a, b]$ with $x_m = mh$ for $m = 0, \ldots, M$, and let $v \in C^4[a, b]$ and $f \in C^2[a, b]$. Suppose $\tilde{V}(x, t)$ is the spline that interpolates the solution curve of this problem at the knots $x_m \in \Pi$. Then there exist constants $\digamma_m$, not depending on h, such that*

$$\|\xi^j\big(v(x,t) - \tilde{V}(x,t)\big)\|_\infty \leq \digamma_j h^{4-j} \quad \forall\, t \geq 0,\ j = 0, 1, 2. \tag{32}$$

Lemma 4.1. *The extended B-splines in (10) satisfy the inequality*

$$\sum_{m=0}^{M} |S_m(x, \lambda)| \leq 1.75 \quad \textit{for } 0 \leq x \leq 1. \tag{33}$$

Proof: By the triangle inequality we have

$$\left|\sum_{m=0}^{M} S_m(x, \lambda)\right| \leq \sum_{m=0}^{M} |S_m(x, \lambda)|.$$

For any knot x_m, we have

$$\begin{aligned}\sum_{m=0}^{M} |S_m(x, \lambda)| &= |S_{m-1}(x_m, \lambda)| + |S_m(x_m, \lambda)|\\ &+ |S_{m+1}(x_m, \lambda)| = 1 < \frac{7}{4}.\end{aligned}$$

From (11) we obtain

$$\begin{aligned}S_m(x_m, \lambda) &= \frac{1}{12}(8+\lambda), \quad S_{m-1}(x_{m-1}, \lambda) = \frac{1}{12}(8+\lambda),\\ S_{m+1}(x_m, \lambda) &= \frac{1}{24}(4-\lambda), \quad S_{m-2}(x_{m-1}, \lambda) = \frac{1}{24}(4-\lambda).\end{aligned}$$

Then, for $x \in [x_{m-1}, x_m]$, $S_m(x, \lambda)$ and $S_{m-1}(x, \lambda)$ are bounded above by $\frac{1}{12}(8 + \lambda)$.

Similarly, $S_{m+1}(x, \lambda)$ and $S_{m-2}(x, \lambda)$ are bounded above by $\frac{1}{24}(4 - \lambda)$

TABLE 3 | Comparison of absolute errors for Example 5.1 using three different methods with $M = 100$, $\Delta t = 0.001$, and $\alpha = 1.4$ or 1.6.

α	(x, t)	VIM [34]	SCCM [42]	Proposed method
	(0.1, 0.1)	9.2852×10^{-3}	8.4385×10^{-4}	3.6460×10^{-7}
	(0.2, 0.2)	2.2201×10^{-3}	1.1433×10^{-4}	3.0191×10^{-7}
	(0.3, 0.3)	3.5651×10^{-2}	5.3780×10^{-3}	1.1558×10^{-6}
	(0.4, 0.4)	4.9628×10^{-2}	1.5545×10^{-4}	1.6174×10^{-5}
1.4	(0.5, 0.5)	6.4449×10^{-2}	5.3227×10^{-4}	8.4214×10^{-6}
	(0.6, 0.6)	7.9514×10^{-2}	1.3268×10^{-3}	6.5725×10^{-6}
	(0.7, 0.7)	9.1443×10^{-2}	1.9159×10^{-3}	3.6215×10^{-6}
	(0.8, 0.8)	8.7942×10^{-2}	2.0414×10^{-3}	3.5112×10^{-6}
	(0.9, 0.9)	9.2321×10^{-4}	1.8996×10^{-3}	5.7354×10^{-8}
	(0.1, 0.1)	4.1518×10^{-4}	1.1685×10^{-4}	7.3256×10^{-6}
	(0.2, 0.2)	1.0319×10^{-3}	2.5887×10^{-4}	2.3576×10^{-5}
	(0.3, 0.3)	1.7757×10^{-2}	2.8863×10^{-5}	2.1107×10^{-5}
	(0.4, 0.4)	2.6987×10^{-2}	2.3912×10^{-4}	1.6174×10^{-5}
1.6	(0.5, 0.5)	3.8327×10^{-2}	1.7692×10^{-5}	8.3440×10^{-6}
	(0.6, 0.6)	5.0993×10^{-2}	1.4174×10^{-4}	6.9744×10^{-7}
	(0.7, 0.7)	6.1379×10^{-2}	1.4334×10^{-5}	3.5898×10^{-6}
	(0.8, 0.8)	5.6577×10^{-2}	1.6653×10^{-4}	3.5003×10^{-6}
	(0.9, 0.9)	3.8618×10^{-2}	1.7449×10^{-5}	5.5205×10^{-8}

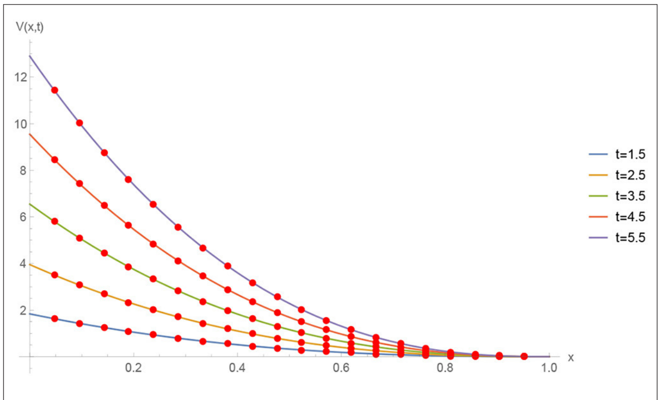

FIGURE 1 | Numerical solution of Example 5.1 with $\Delta t = 0.001$, $M = 100$, and $\alpha = 1.5$ at different time stages.

For any point $x_{m-1} \le x \le x_m$, we obtain

$$\sum_{m=0}^{M} |S_m(x,\lambda)| = |S_{m-1}(x,\lambda)| + |S_m(x,\lambda)| + |S_{m+1}(x,\lambda)| + |S_{m-2}(x,\lambda)| = \frac{1}{12}(\lambda+20).$$

Since $\lambda \in [-8,1]$, we have $1 \le \frac{5}{3} + \lambda \le 1.75$. Hence,

$$\sum_{m=0}^{M} |S_m(x,\lambda)| \le 1.75.$$

Theorem 3. *The extended cubic B-spline approximation $V(x,t)$ for the analytical exact solution $v(x,t)$ of problem (1)–(3) exists, and if $f \in C^2[0,1]$ then*

$$\|v(x,t) - V(x,t)\|_\infty \le \widetilde{F} h^2 \quad \forall\, t \ge 0, \tag{34}$$

where h is reasonably small and $\widetilde{F} > 0$ is a constant not depending on h.

Proof: Let $\tilde{V}(x,t) = \sum_{m=0}^{M} d_m(t)\eta_m(x)$ be the calculated spline for the approximate solution $V(x,t)$ and the exact solution $v(x,t)$.

Let $Lv(x_m,t) = LV(x_m,t) = \tilde{y}(x_m,t)$, with $m = 0:1:M$, be the collocation conditions. Then

$$L\tilde{V}(x,t) = \tilde{y}(x_m,t), \quad m = 0:1:M.$$

Now, at any time step, the problem can be expressed in the form of a difference equation $L(\tilde{V}(x_m,t) - V(x_m,t))$ as

$$\begin{aligned}
&(\beta b_1 + \rho_1 b_1 + \rho b_4)\zeta_{m-1}^{r+1} + (\beta b_2 + \rho_1 b_2 + \rho b_5)\zeta_m^{r+1} \\
&+ (\beta b_1 + \rho_1 b_1 + \rho b_4)\zeta_{m+1}^{r+1} \\
&= 2\beta\big(b_1\zeta_{m-1}^{r} + b_2\zeta_m^{r} + b_1\zeta_{m+1}^{r}\big) - \beta\big(b_1\zeta_{m-1}^{r-1} + b_2\zeta_m^{r-1} \\
&\quad + b_1\zeta_{m+1}^{r-1}\big) - \beta\sum_{k=1}^{r} p_k\Big[b_1\big(\zeta_{m-1}^{r-k+1} - 2\zeta_{m-1}^{r-k} + \zeta_{m-1}^{r-k-1}\big) \\
&+ b_2\big(\zeta_m^{r-k+1} - 2\zeta_m^{r-k} + \zeta_m^{r-k-1}\big) \\
&\quad + b_1\big(\zeta_{m+1}^{r-k+1} - 2\zeta_{m+1}^{r-k} + \zeta_{m+1}^{r-k-1}\big)\Big] + \frac{1}{h^2}\eta_m^{r+1}.
\end{aligned} \tag{35}$$

The boundary conditions can be rewritten as

$$b_1\zeta_{m-1}^{r+1} + b_2\zeta_m^{r+1} + b_1\zeta_{m+1}^{r+1} = 0, \quad m = 0, M,$$

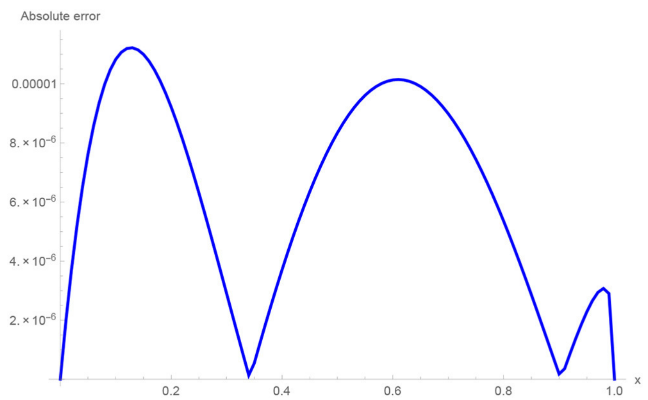

FIGURE 3 | Absolute error for Example 5.1 when $M = 100$, $\alpha = 1.50$, and $\Delta t = 0.001$.

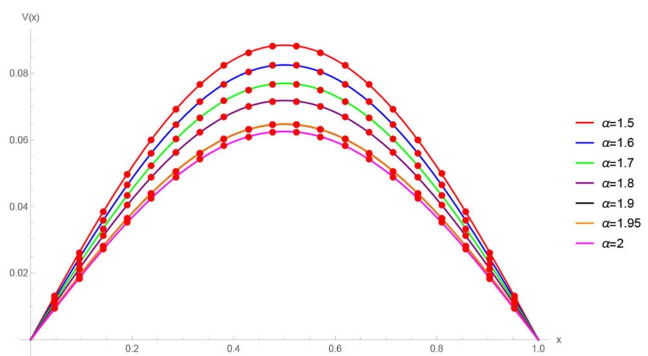

FIGURE 4 | Approximate solution of Example 5.1 with $M = 100$, $t = 0.5$, and different values of α.

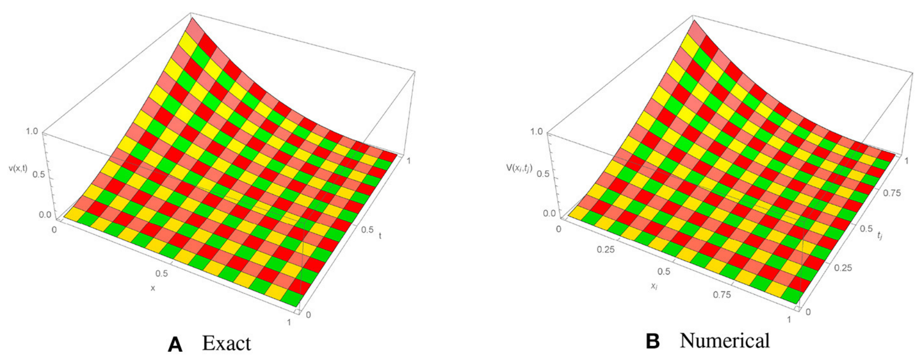

FIGURE 2 | Exact and approximate solutions of Example 5.1 with $M = 100$, $\Delta t = 0.001$, and $\alpha = 1.50$. **(A)** Exact. **(B)** Numerical.

where

$$\zeta_m^r = \xi_m^r - d_m^r, \quad m = 0:1:M,$$

and

$$\eta_m^r = h^2[y_m^r - \tilde{y}_m^r], \quad m = 0:1:M.$$

From (32) we have

$$|\eta_m^r| = h^2|y_m^r - \tilde{y}_m^r| \le \digamma h^4.$$

We define $\eta^r = \max\{|\eta_m^r| : 0 \le m \le M\}$, $\tilde{e}_m^r = |\zeta_m^r|$ and $\tilde{e}^r = \max\{|e_m^r| : 0 \le m \le M\}$.

For $r = 0$, Equation (35) transforms into the following relation:

$$\begin{aligned}&(\beta b_1 + \rho_1 b_1 + \rho b_4)\zeta_{m-1}^1 + (\beta b_2 + \rho_1 b_2 + \rho b_5)\zeta_m^1 \\ &+ (\beta b_1 + \rho_1 b_1 + \rho b_4)\zeta_{m+1}^1 \\ &= (\beta + \rho_1)\big(b_1\zeta_{m-1}^0 + b_2\zeta_m^0 + b_1\zeta_{m+1}^0\big) + \frac{1}{h^2}\eta_m^1.\end{aligned}$$

Using the initial condition $e^0 = 0$, we obtain

$$\begin{aligned}&(\beta b_2 + \rho_1 b_2 + \rho b_5)\zeta_m^1 = (\beta b_1 + \rho b_4)(\zeta_{m+1}^1 - \zeta_{m-1}^1) \\ &+\rho_1 b_1(\zeta_{m+1}^1 - \zeta_{m-1}^1) + \frac{1}{h^2}\eta_m^1.\end{aligned}$$

Taking absolute values of η_m^r and ζ_m^r and with adequately small h, we have

$$\tilde{e}_m^1 \le \frac{6\digamma h^4}{\beta h^2(\lambda+2) + 12(-2-\lambda)\rho + \rho_1 h^2(2+\lambda)}$$

using the boundary conditions, from which we conclude that

$$\tilde{e}^1 \le \digamma_1 h^2, \tag{36}$$

where $\digamma_1$ is independent of the spatial grid spacing.

Using the induction technique, we assume that $\tilde{e}_m^k \le \digamma_k h^2$ is true for $k = 1:1:r$.

Let $\digamma = \max\{\digamma_k : 0 \le k \le r\}$; then Equation (35) becomes

$$\begin{aligned}&(\beta b_1 + \rho_1 b_1 + \rho b_4)\zeta_{m-1}^{r+1} + (\beta b_2 + \rho_1 b_2 + \rho b_5)\zeta_m^{r+1} \\ &+ (\beta b_1 + \rho_1 b_1 + \rho b_4)\zeta_{m+1}^{r+1} \\ &= 2\beta\big(b_1\zeta_{m-1}^r + b_2\zeta_m^r + b_1\zeta_{m+1}^r\big) - \beta\big(b_1\zeta_{m-1}^{r-1} + b_2\zeta_m^{r-1} + b_1\zeta_{m+1}^{r-1}\big) \\ &\quad + \beta\Big[(p_0 - 2p_1 + p_2)(b_1\zeta_{m-1}^r + b_2\zeta_m^r + b_1\zeta_{m+1}^r) \\ &+ (p_1 - 2p_2 + p_3)(b_1\zeta_{m-1}^{r-1} + b_2\zeta_m^{r-1} + b_1\zeta_{m+1}^{r-1}) \\ &\qquad + \cdots + (p_{r-4} - 2p_{r-3} + p_{r-2})(b_1\zeta_{m-1}^1 + b_2\zeta_m^1 \\ &+ b_1\zeta_{m+1}^1) + p_{r-1}(b_1\zeta_{m-1}^0 + b_2\zeta_m^0 + b_1\zeta_{m+1}^0)\Big] \quad + \frac{1}{h^2}\eta_m^{r+1}.\end{aligned}$$

Again, taking absolute values of η_m^r and ζ_m^r, we have

$$\begin{aligned}\tilde{e}_m^{r+1} \le &\frac{6\digamma h^2}{\beta h^2(2+\lambda) + 12(-2-\lambda)\rho + \rho_1 h^2(2+\lambda)} \\ &\Big[2\beta(b_1\zeta_{m-1}^r + b_2\zeta_m^r + b_1\zeta_m^r) \\ &\quad - \beta\sum_{k=0}^{r-1}(p_{k+1} - 2p_k - p_{k-1})\digamma h^2 + \digamma h^2\Big].\end{aligned}$$

TABLE 4 | Experimental order of convergence (EOC) for Example 5.1 with $\alpha = 1.3$ and $\Delta t = 0.001$.

M	L_∞	EOC	L_2	EOC
10	3.1950×10^{-2}	—	2.9355×10^{-2}	—
20	9.0451×10^{-3}	1.8206	8.7109×10^{-3}	1.7527
40	2.4778×10^{-3}	1.8680	2.2128×10^{-3}	1.9769
80	6.3842×10^{-4}	1.9564	5.9376×10^{-4}	1.8979

TABLE 6 | Absolute and relative errors for Example 5.2 when $M = 100$, $\Delta t = 0.001$ and $\alpha = 1.6$.

		SCCM [42]		Proposed method	
t	x	L_∞	L_2	L_∞	L_2
	0.4	3.1780×10^{-6}	9.0475×10^{-5}	1.1769×10^{-7}	9.1321×10^{-8}
0.4	0.6	3.1780×10^{-6}	9.0475×10^{-5}	1.0126×10^{-6}	8.0341×10^{-7}
	0.8	2.1040×10^{-5}	9.6921×10^{-4}	7.2740×10^{-6}	1.2573×10^{-6}
	0.4	5.8118×10^{-4}	7.6534×10^{-4}	1.8278×10^{-5}	8.9616×10^{-6}
0.8	0.6	2.4754×10^{-4}	5.8118×10^{-4}	1.2788×10^{-6}	7.8014×10^{-7}
	0.8	4.7365×10^{-4}	1.7994×10^{-3}	1.0951×10^{-5}	9.5597×10^{-6}

TABLE 5 | Absolute errors for Example 5.2 when $M = 100$, $\Delta t = 0.001$ using different values of α.

	SCCM [42]			Proposed method		
x	$\alpha = 1.5$	$\alpha = 1.7$	$\alpha = 1.9$	$\alpha = 1.5$	$\alpha = 1.7$	$\alpha = 1.9$
0.1	1.6396×10^{-3}	1.5471×10^{-3}	1.4380×10^{-3}	2.6129×10^{-6}	8.4422×10^{-6}	9.8439×10^{-6}
0.2	1.2808×10^{-3}	1.1272×10^{-3}	9.4914×10^{-4}	3.0564×10^{-5}	1.4959×10^{-7}	6.7965×10^{-6}
0.3	1.0869×10^{-3}	8.9663×10^{-4}	6.7913×10^{-4}	9.7609×10^{-6}	$2,7610 \times 10^{-6}$	1.0853×10^{-5}
0.4	8.4196×10^{-4}	6.3348×10^{-4}	3.9687×10^{-4}	1.9015×10^{-6}	5.8360×10^{-6}	7.0990×10^{-6}
0.5	7.8252×10^{-4}	5.6868×10^{-4}	3.2651×10^{-4}	3.2181×10^{-6}	7.1727×10^{-6}	3.1898×10^{-5}
0.6	8.4196×10^{-4}	6.3348×10^{-4}	3.9687×10^{-4}	1.9015×10^{-5}	5.8360×10^{-6}	4.1207×10^{-6}
0.7	1.0869×10^{-3}	8.9663×10^{-4}	6.7913×10^{-4}	9.7609×10^{-6}	2.7610×10^{-6}	8.6781×10^{-6}
0.8	1.2808×10^{-3}	1.1272×10^{-3}	9.4914×10^{-4}	3.0564×10^{-5}	1.4959×10^{-7}	6.7965×10^{-6}
0.9	1.6396×10^{-3}	1.5471×10^{-3}	1.4380×10^{-3}	2.6129×10^{-6}	8.4422×10^{-6}	9.8439×10^{-6}

Using the boundary conditions, we have

$$\tilde{e}_m^{r+1} \leq \digamma h^2.$$

Hence, for all values of n,

$$\tilde{e}_m^{r+1} \leq \digamma h^2. \tag{37}$$

Now,

$$\tilde{V}(x,t) - V(x,t) = \sum_{m=0}^{M} \big(d_m(t) - \xi_m(t)\big) S_m(x).$$

Taking the infinity norm and applying Lemma (3.1), we obtain

$$\|\tilde{V}(x,t) - V(x,t)\|_\infty \leq 1.75 \digamma h^2. \tag{38}$$

Making use of the triangle inequality, we get

$$\|v(x,t) - V(x,t)\|_\infty \leq \|v(x,t) - \tilde{V}(x,t)\|_\infty + \|\tilde{V}(x,t) - V(x,t)\|_\infty. \tag{39}$$

Using the inequalities (32) and (38) in (39), we obtain

$$\|v(x,t) - V(x,t)\|_\infty \leq \digamma_0 h^4 + 1.75 \digamma h^2 = \widetilde{\digamma} h^2,$$

where $\widetilde{\digamma} = \digamma_0 h^2 + 1.75 \digamma$.

Using the above theorem with expression (5), it is easy to conclude that the numerical approach converges unconditionally. Therefore,

$$\|v(x,t) - V(x,t)\|_\infty \leq \widetilde{\digamma} h^2 + \psi(\Delta t)^{2-\alpha},$$

where $\widetilde{\digamma}$ is a constant and $\alpha \in (1,2]$. Hence, theoretically, the proposed scheme is $O(h^2 + \Delta t^{2-\alpha})$ accurate.

5. NUMERICAL RESULTS AND DISCUSSION

To examine the accuracy of the proposed method, we conduct a numerical study of some test problems. The L_∞ and L_2 error norms are calculated as [53]

$$L_\infty = \max_{0 \leq m \leq M} |V(x_m,t) - v(x_m,t)|,$$

$$L_2 = \sqrt{h \sum_{m=0}^{M} |V(x_m,t) - v(x_m,t)|^2}.$$

Also, the experimental order of convergence (EOC) is computed by the following important formula [54]:

$$\text{EOC} = \frac{1}{\log 2} \log\left[\frac{L_\infty(2m)}{L_\infty(m)}\right].$$

All numerical computations were performed using `Mathematica` 9.0.

Example 5.1. Consider the non-linear time-fractional KGE [42]

$$\frac{\partial^\alpha v}{\partial t^\alpha} - \frac{\partial^2 v}{\partial x^2} + v^2(x,t) = f(x,t), \qquad 0 < t \leq 1,\ 0 < x \leq 1, \tag{40}$$

TABLE 7 | Absolute errors for Example 5.2 when $M = 100$ and $\Delta t = 0.001$.

α	(x,t)	VIM [34]	SCCM [42]	Proposed method
	(0.1, 0.1)	3.9211×10^{-5}	2.3809×10^{-5}	1.9749×10^{-6}
	(0.2, 0.2)	6.1713×10^{-4}	5.2644×10^{-5}	1.7326×10^{-5}
	(0.3, 0.3)	2.1989×10^{-3}	6.0187×10^{-6}	5.2839×10^{-6}
	(0.4, 0.4)	2.5545×10^{-3}	6.6640×10^{-5}	9.9062×10^{-6}
1.4	(0.5, 0.5)	5.3405×10^{-3}	4.0011×10^{-5}	1.3396×10^{-6}
	(0.6, 0.6)	3.1409×10^{-2}	1.5837×10^{-4}	1.3557×10^{-5}
	(0.7, 0.7)	8.0092×10^{-2}	9.1922×10^{-4}	9.6832×10^{-6}
	(0.8, 0.8)	1.3528×10^{-1}	2.9084×10^{-3}	3.5290×10^{-5}
	(0.9, 0.9)	1.4272×10^{-1}	3.8732×10^{-3}	9.0059×10^{-6}
	(0.1, 0.1)	1.0402×10^{-5}	2.3809×10^{-5}	1.4963×10^{-6}
	(0.2, 0.2)	1.4424×10^{-4}	5.2644×10^{-5}	1.5765×10^{-6}
	(0.3, 0.3)	6.7115×10^{-5}	6.0187×10^{-6}	2.1699×10^{-7}
	(0.4, 0.4)	3.0493×10^{-3}	6.4440×10^{-5}	1.1769×10^{-6}
1.6	(0.5, 0.5)	1.6350×10^{-2}	4.0011×10^{-5}	1.2375×10^{-6}
	(0.6, 0.6)	4.9599×10^{-2}	1.5837×10^{-4}	2.1232×10^{-6}
	(0.7, 0.7)	1.0675×10^{-1}	9.1922×10^{-4}	1.8721×10^{-6}
	(0.8, 0.8)	1.6942×10^{-1}	2.9084×10^{-3}	1.0951×10^{-5}
	(0.9, 0.9)	1.7521×10^{-1}	3.8732×10^{-3}	2.2989×10^{-5}

TABLE 8 | Experimental order of convergence (EOC) for Example 5.2 with $\alpha = 1.5$ and $\Delta t = 0.001$.

M	L_∞	EOC	L_2	EOC
10	2.0835×10^{-2}	–	1.8459×10^{-2}	–
20	5.2813×10^{-3}	1.9760	4.7833×10^{-3}	1.9482
40	1.3057×10^{-3}	2.0161	1.1406×10^{-3}	2.0688
80	3.2509×10^{-4}	2.0059	2.8172×10^{-4}	2.0174

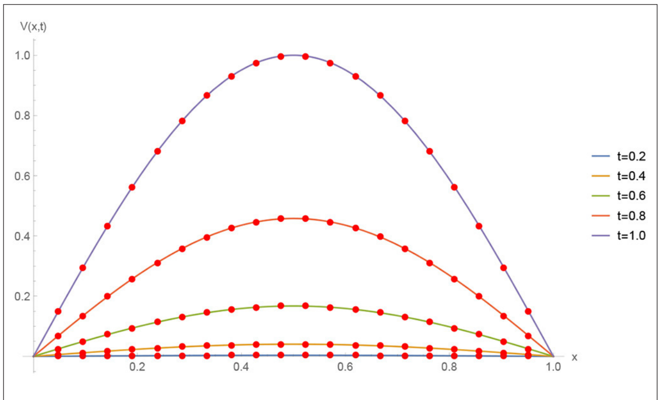

FIGURE 5 | Numerical solution for Example 5.2 with $\Delta t = 0.001$, $M = 100$, and $\alpha = 1.5$ at different time stages.

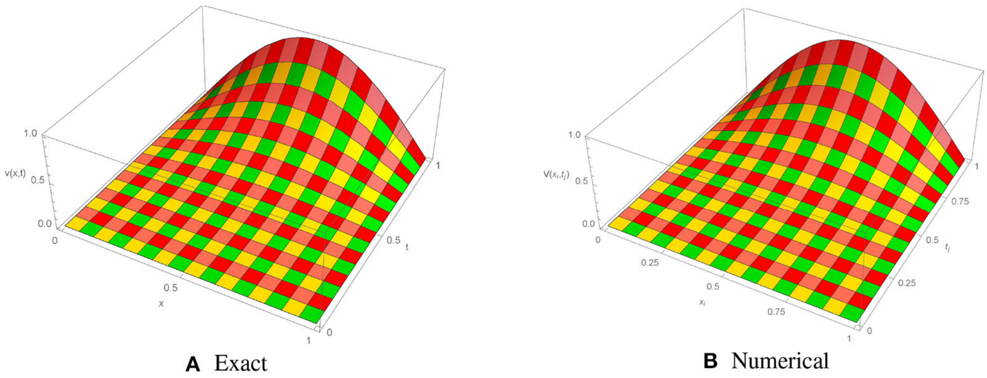

FIGURE 6 | Exact and numerical solutions of Example 5.2 with $M = 100$, $\Delta t = 0.001$, and $\alpha = 1.5$. **(A)** Exact. **(B)** Numerical.

where $f(x,t) = \frac{\Gamma(\frac{5}{2})}{\Gamma(\frac{5}{2}-\alpha)}(1-x)^{\frac{5}{2}}t^{\frac{3}{2}-\alpha} - \frac{15}{4}(1-x)^{\frac{1}{2}}t^{\frac{3}{2}} + (1-x)^5 t^3$. The initial/end conditions can be extracted from the analytical exact solution $(1-x)^{\frac{5}{2}}t^{\frac{3}{2}-\alpha}$.

For Example 5.1, the piecewise-defined approximate solution obtained using the proposed method with $\alpha = 1.25$, $0 \leq x \leq 1$, $n = 100$, and $\Delta t = 0.01$ is given by

$$V(x) = \begin{cases} 0. + x(297.276 + x(-29930.4 + x(993222. + 225927.x))) & \text{if } x \in [0.00, 0.01], \\ 0.999999 + x(-2.49738 + x(1.82587 + (1.38305 - 27.8749x)x)) & \text{if } x \in [0.01, 0.02], \\ 0.99999 + x(-2.49605 + x(1.75961 + (2.48215 - 27.7432x)x)) & \text{if } x \in [0.02, 0.03], \\ 0.99996 + x(-2.49308 + x(1.66094 + (3.57055 - 27.6103x)x)) & \text{if } x \in [0.03, 0.04], \\ \vdots & \vdots \\ -0.118298 + x(6.72761 + x(-26.6775 + (38.9565 - 20.3042x)x)) & \text{if } x \in [0.49, 0.50], \\ -0.201484 + x(7.21369 + x(-27.5747 + (39.3734 - 20.1068x)x)) & \text{if } x \in [0.50, 0.51], \\ \vdots & \vdots \\ -2.7339 + x(13.6165 + x(-24.3154 + (18.715 - 5.28228x)x)) & \text{if } x \in [0.96, 0.97], \\ -1.89304 + x(10.2593 + x(-19.2941 + (15.3811 - 4.45319x)x)) & \text{if } x \in [0.97, 0.98], \\ -0.518579 + x(5.07656 + x(-12.0155 + (10.8746 - 3.41708x)x)) & \text{if } x \in [0.98, 0.99], \\ 4.86293 + x(-13.1733 + x(10.3424 + (-0.616646 - 1.41541x)x)) & \text{if } x \in [0.99, 1.00]. \end{cases}$$

The absolute numerical errors at different grid points of the RECBS solution for Example 5.1 using $\Delta t = 0.001$ and $M = 100$ are reported in **Table 1**. It can easily be seen that our scheme is more accurate than the SCCM [42]. In **Table 2** the absolute and relative numerical errors are listed for our method with $M = 100$, $\Delta t = 0.001$, and $\alpha = 1.6$ at $x = 0.4, 0.6, 0.8$ when $t = 0.4, 0.8$. We can see that the computational results are superior to those obtained from the SCCM [42]. **Table 3** compares the absolute errors of the proposed method, the variational iteration method (VIM) [34], and the SCCM [42] under different values of α. **Figure 1** shows the behavior at different time stages of numerical solutions obtained using $\alpha = 1.5$, $M = 100$, and $\Delta t = 0.001$. The 3D visuals of exact and numerical solutions with $\alpha = 1.5$ and $M = 100$ are shown in **Figure 2**. The comparison between the exact and approximate solutions using $M = 100$ is plotted in **Figure 3**. **Figure 4** depicts the absolute error between the exact and numerical solutions when $\alpha = 1.3$, $M = 100$, and $\Delta t = 0.001$. The values of the EOC along the spatial grid, using $\Delta t = 0.001$ and $\alpha = 1.5$, are given in **Table 4**. The experimental rate of convergence of the proposed method is found to be in line with the theoretical results.

Example 5.2. Consider the fractional KGE [34, 42]

$$\frac{\partial^\alpha}{\partial t^\alpha}v(x,t) - \frac{\partial^2}{\partial x^2}v(x,t) + v(x,t) + \frac{3}{2}v^3(x,t) = f(x,t), \quad 0 < x \leq 1, \ 0 < t \leq 1, \quad (41)$$

where the forcing term $f(x,t)$ on right-hand side is given by

$$f(x,t) = \frac{1}{2}\Gamma(3+\alpha)\sin(\pi x)t^2 + (1+\pi^2)t^{2+\alpha}\sin(\pi x) + \frac{3}{2}\left[\sin(\pi x)t^{2+\alpha}\right]^3,$$

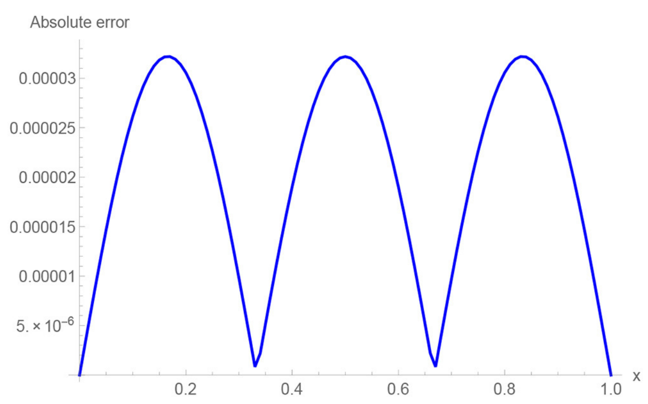

FIGURE 7 | Absolute error for Example 5.2 when $M = 100$, $\alpha = 1.5$, and $\Delta t = 0.001$.

For Example 5.2, the piecewise-defined numerical solution obtained using the proposed method with $\alpha = 1.5$, $0 \le x \le 1$, $n = 100$, and $\Delta t = 0.01$ is given by

$$V(x) = \begin{cases} 8.71156 \times 10^{-19} + x(3.13867 + x(2.8549 \times 10^{-}14 + (-4.97167 - 11.4015x)x)) & \text{if } x \in [0.00, 0.01], \\ -1.14461 \times 10^{-6} + x(3.13904 + x(-0.041176 + (-3.14329 - 34.194x)x)) & \text{if } x \in [0.01, 0.02], \\ -0.0000194466 + x(3.14196 + x(-0.205754 + (0.51013 - 56.9551x)x)) & \text{if } x \in [0.02, 0.03], \\ -0.000112001 + x(3.15183 + x(-0.575584 + (5.98188 - 79.6639x)x)) & \text{if } x \in [0.03, 0.04], \\ \vdots & \vdots \\ -40.7681 + x(339.328 + x(-1039.38 + (1422.21 - 733.23x)x)) & \text{if } x \in [0.49, 0.50], \\ -44.2829 + x(360.934 + x(-1083.83 + (1453.18 - 733.97x)x)) & \text{if } x \in [0.50, 0.51], \\ \vdots & \vdots \\ -71.1059 + x(298.709 + x(-460.613 + (312.674 - 79.6639x)x)) & \text{if } x \in [0.96, 0.97], \\ -53.5088 + x(223.56 + x(-340.406 + (227.31 - 56.9551x)x)) & \text{if } x \in [0.97, 0.98], \\ -34.2394 + x(143.149 + x(-214.635 + (139.919 - 34.194x)x)) & \text{if } x \in [0.98, 0.99], \\ -13.2345 + x(57.3823 + x(-83.3239 + (50.5776 - 11.4015x)x)) & \text{if } x \in [0.99, 1.00]. \end{cases}$$

The initial/boundary conditions can be extracted from the analytical exact solution $v(x,t) = \sin(\pi x)t^{2+\alpha}$. The absolute numerical errors at different grid points of the RECBS solution for Example 5.2 using $\Delta t = 0.001$ and $M = 100$ are listed in **Table 5**. Again it can be observed that our scheme is more accurate than the SCCM [42]. **Table 6** reports the absolute and relative errors in our numerical computation with $M = 100$, $\Delta t = 0.001$, and $\alpha = 1.6$ at $x = 0.4, 0.6, 0.8$ when $t = 0.4, 0.8$. It is clear that the results are better than those obtained by the SCCM [42]. **Table 7** compares the absolute errors of the proposed method, VIM [34], and SCCM [42] under different values of α.

The EOC in the spatial direction, using $\Delta t = 0.001$ and $\alpha = 1.50$, is tabulated in **Table 8**. The experimental rate of convergence of the proposed scheme is found to be in line with the theoretical prediction. **Figure 5** shows the behavior at different time stages of numerical solutions obtained using $\alpha = 1.5$, $M = 100$, and $\Delta t = 0.001$. The 3D plots of exact and numerical solutions with $\alpha = 1.5$ and $M = 100$ are displayed in **Figure 6**. The absolute error between the exact and approximate solutions using $\alpha = 1.3$, $M = 100$, and $\Delta t = 0.001$ is plotted in **Figure 7**.

6. CONCLUSION

In this work we have conducted a numerical investigation of the time-fractional Klein–Gordon equation by applying the redefined extended cubic B-spline collocation method. A finite central difference formulation is employed for temporal discretization, while a set of redefined extended cubic B-spline functions is used to interpolate the solution curve in the spatial direction. The unconditional stability of the proposed scheme is established, and the orders of convergence along the space and time grids are shown to be $O(h^2)$ and $O(\Delta t)^{2-\alpha}$, respectively. The computational outcomes of the proposed algorithm show that the order of convergence agrees with the theoretical results. The numerical scheme has been tested on different problems, and comparison of the results reveals our method's advantage over VIM [34] and SCCM [42].

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

REFERENCES

1. Rossikhin YA, Shitikova M. Application of fractional derivatives to the analysis of damped vibrations of viscoelastic single mass systems. *Acta Mech.* (1997) **120**:109–25. doi: 10.1007/BF01174319
2. Rudolf H. *Applications of FractionalCalculus in Physics*. Singapore; River Edge, NJ: World Scientific (2000).
3. Metzler R, Klafter J. The restaurant at the end of the random walk: recent developments in the description of anomalous transport by fractional dynamics. *J Phys A Math Gen.* (2004) **37**:R161. doi: 10.1088/0305-4470/37/31/R01

4. Luo MJ, Milovanovic GV, Agarwal P. Some results on the extended beta and extended hypergeometric functions. *Appl Math Comput.* (2014) **248**:631–51. doi: 10.1016/j.amc.2014.09.110
5. Zhang Y. Time-fractional Klein-Gordon equation: formulation and solution using variational methods. *WSEAS Trans Math*. (2016) **15**:206–14.
6. Ruzhansky M, Cho YJ, Agarwal P, Area I. *Advances in Real and Complex Analysis With Applications*. Singapore: Springer (2017).
7. Owolabi KM, Hammouch Z. Mathematical modeling and analysis of two-variable system with noninteger-order derivative. *Chaos*. (2019) **29**:013145. doi: 10.1063/1.5086909
8. Agarwal P, Baleanu D, Chen Y, Momani S, Machado JAT. Fractional calculus. In: *ICFDA: International Workshop on Advanced Theory and Applications of Fractional Calculus*. Amman (2019). doi: 10.1007/978-981-15-0430-3
9. Babaei A, Jafari H, Ahmadi M. A fractional order HIV/AIDS model based on the effect of screening of unaware infectives. *Math Methods Appl Sci*. (2019) **42**:2334–43. doi: 10.1002/mma.5511
10. Babaei A, Jafari H, Liya A. Mathematical models of HIV/AIDS and drug addiction in prisons. *Eur Phys J Plus*. (2020) **135**:395. doi: 10.1140/epjp/s13360-020-00400-0
11. Yuste SB, Acedo L. An explicit finite difference method and a new von Neumann-type stability analysis for fractional diffusion equations. *SIAM J Numer Anal*. (2005) **42**:1862–74. doi: 10.1137/0306 02666
12. Sweilam N, Nagy A. Numerical solution of fractional wave equation using Crank-Nicholson method. *World Appl Sci J*. (2011) **13**:71–5.
13. Ur-Rehman M, Khan RA. The Legendre wavelet method for solving fractional differential equations. *Commun Nonlin Sci Numer Simul*. (2011) **16**:4163–73. doi: 10.1016/j.cnsns.2011.01.014
14. Bhrawy A, Tharwat M, Yildirim A. A new formula for fractional integrals of Chebyshev polynomials: application for solving multi-term fractional differential equations. *Appl Math Modell*. (2013) **37**:4245–52. doi: 10.1016/j.apm.2012.08.022
15. Rad J, Kazem S, Shaban M, Parand K, Yildirim A. Numerical solution of fractional differential equations with a Tau method based on Legendre and Bernstein polynomials. *Math Methods Appl Sci*. (2014) **37**:329–42. doi: 10.1002/mma.2794
16. Mohebbi A, Abbaszadeh M, Dehghan M. High-order difference scheme for the solution of linear time fractional Klein-Gordon equations. *Numer Methods Partial Differ Equat*. (2014) **30**:1234–53. doi: 10.1002/num.21867
17. Heydari M, Hooshmandasl M, Mohammadi F, Cattani C. Wavelets method for solving systems of nonlinear singular fractional Volterra integro-differential equations. *Commun Nonlin Sci Numer Simul*. (2014) **19**:37–48. doi: 10.1016/j.cnsns.2013.04.026
18. Salahshour S, Ahmadian A, Senu N, Baleanu D, Agarwal P. On analytical solutions of the fractional differential equation with uncertainty: application to the basset problem. *Entropy*. (2015) **17**:885–902. doi: 10.3390/e17020885
19. Sweilam N, Nagy A, El-Sayed AA. Second kind shifted Chebyshev polynomials for solving space fractional order diffusion equation. *Chaos Solit Fract*. (2015) **73**:141–7. doi: 10.1016/j.chaos.2015.01.010
20. Ramezani M, Jafari H, Johnston SJ, Baleanu D. Complex b-spline collocation method for solving weakly singular Volterra integral equations of the second kind. *Miskolc Math Notes*. (2015) **16**:1091–103. doi: 10.18514/MMN.2015.1469
21. Badr M, Yazdani A, Jafari H. Stability of a finite volume element method for the time-fractional advection-diffusion equation. *Numer Methods Partial Differ Equat.* Wiley online Library. (2018) **34**:1459–71. doi: 10.1002/num.22243
22. Agarwal P, Agarwal RP, Ruzhansky M. *Special Functions and Analysis of Differential Equations*. (2020).
23. Atangana A, Secer A. A note on fractional order derivatives and table of fractional derivatives of some special functions. *Abstr Appl Anal.* (2013) **2013**:279681. doi: 10.1155/2013/279681
24. Srivastava H, Agarwal P. Certain fractional integral operators and the generalized incomplete hypergeometric functions. *Appl Appl Math.* (2013) **8**:333–45.
25. Goufo EFD. Solvability of chaotic fractional systems with 3D four-scroll attractors. *Chaos Solit Fract*. (2017) **104**:443–51. doi: 10.1016/j.chaos.2017.08.038
26. Asif N, Hammouch Z, Riaz M, Bulut H. Analytical solution of a Maxwell fluid with slip effects in view of the Caputo-Fabrizio derivative. *Eur Phys J Plus*. (2018) **133**:272. doi: 10.1140/epjp/i2018-12098-6
27. Singh J, Kumar D, Hammouch Z, Atangana A. A fractional epidemiological model for computer viruses pertaining to a new fractional derivative. *Appl Math Comput*. (2018) **316**:504–15. doi: 10.1016/j.amc.2017.08.048
28. Atangana A, Goufo EFD. Conservatory of Kaup-Kupershmidt equation to the concept of fractional derivative with and without singular kernel. *Acta Math Appl Sin*. (2018) **34**:351–61. doi: 10.1007/s10255-018-0757-7
29. Owolabi KM, Atangana A. Computational study of multi-species fractional reaction-diffusion system with ABC operator. *Chaos Solit Fract*. (2019) **128**:280–9. doi: 10.1016/j.chaos.2019.07.050
30. Uçar S, Uçar E, Özdemir N, Hammouch Z. Mathematical analysis and numerical simulation for a smoking model with Atangana-Baleanu derivative. *Chaos Solit Fract*. (2019) **118**:300–6. doi: 10.1016/j.chaos.2018.12.003
31. Goufo EFD, Toudjeu IT. Analysis of recent fractional evolution equations and applications. *Chaos Solit Fract*. (2019) **126**:337–50. doi: 10.1016/j.chaos.2019.07.016
32. Batiha B, Noorani MSM, Hashim I. Numerical solution of sine-Gordon equation by variational iteration method. *Phys Lett A*. (2007) **370**:437–40. doi: 10.1016/j.physleta.2007.05.087
33. Yusufoğlu E. The variational iteration method for studying the Klein-Gordon equation. *Appl Math Lett*. (2008) **21**:669–74. doi: 10.1016/j.aml.2007.07.023
34. Odibat Z, Momani S. The variational iteration method: an efficient scheme for handling fractional partial differential equations in fluid mechanics. *Comput Math Appl*. (2009) **58**:2199–208. doi: 10.1016/j.camwa.2009.03.009
35. Jafari H, Saeidy M, Arab Firoozjaee M. Solving nonlinear Klein-Gordon equation with a quadratic nonlinear term using homotopy analysis method. *Iran J Optimiz*. (2010) **2**:130–38.
36. Jafari H, Khalique CM, Ramezani M, Tajadodi H. Numerical solution of fractional differential equations by using fractional B-spline. *Central Eur J Phys*. (2013) **11**:1372–6. doi: 10.2478/s11534-013-0222-4
37. Vong S, Wang Z. A compact difference scheme for a two dimensional fractional Klein-Gordon equation with Neumann boundary conditions. *J Comput Phys*. (2014) **274**:268–82. doi: 10.1016/j.jcp.2014.06.022
38. Vong S, Wang Z. A high-order compact scheme for the nonlinear fractional Klein-Gordon equation. *Numer Methods Partial Differ Equat.* (2015) **31**:706–22. doi: 10.1002/num.21912
39. Dehghan M, Abbaszadeh M, Mohebbi A. An implicit RBF meshless approach for solving the time fractional nonlinear sine-Gordon and Klein-Gordon equations. *Eng Anal Bound Elements*. (2015) **50**:412–34. doi: 10.1016/j.enganabound.2014.09.008
40. Jafari H. Numerical solution of time-fractional Klein-Gordon equation by using the decomposition methods. *J Comput Nonlin Dyn*. (2016) **11**:041015. doi: 10.1115/1.4032767
41. Chen H, Lü S, Chen W, et al. A fully discrete spectral method for the nonlinear time fractional Klein-Gordon equation. *Taiwan J Math*. (2017) **21**:231–51. doi: 10.11650/tjm.21.2017.7357
42. Nagy A. Numerical solution of time fractional nonlinear Klein-Gordon equation using Sinc-Chebyshev collocation method. *Appl Math Comput*. (2017) **310**:139–48. doi: 10.1016/j.amc.2017.04.021
43. Kanwal A, Phang C, Iqbal U. Numerical solution of fractional diffusion wave equation and fractional Klein-Gordon equation via two-dimensional Genocchi polynomials with a Ritz-Galerkin method. *Computation*. (2018) **6**:40. doi: 10.3390/computation6030040
44. Lyu P, Vong S. A linearized second-order scheme for nonlinear time fractional Klein-Gordon type equations. *Numer Algorithms*. (2018) **78**:485–511. doi: 10.1007/s11075-017-0385-y
45. Doha E, Abdelkawy M, Amin A, Lopes AM. A space-time spectral approximation for solving nonlinear variable-order fractional sine and Klein-Gordon differential equations. *Comput Appl Math*. (2018) **37**:6212–29. doi: 10.1007/s40314-018-0695-2
46. Amin M, Abbas M, Iqbal MK, Ismail AIM, Baleanu D. A fourth order non-polynomial quintic spline collocation technique for solving time fractional superdiffusion equations. *Adv Differ Equat.* (2019) **2019**:1–21. doi: 10.1186/s13662-019-2442-4
47. Khalid N, Abbas M, Iqbal MK, Baleanu D. A numerical algorithm based on modified extended B-spline functions for solving time-fractional diffusion

wave equation involving reaction and damping terms. *Adv Differ Equat*. (2019) **2019**:378. doi: 10.1186/s13662-019-2318-7

48. Wasim I, Abbas M, Iqbal M. A new extended B-spline approximation technique for second order singular boundary value problems arising in physiology. *J Math Comput Sci*. (2019) **19**:258–67. doi: 10.22436/jmcs.019.04.06
49. Sharifi S, Rashidinia J. Numerical solution of hyperbolic telegraph equation by cubic B-spline collocation method. *Appl Math Comput*. (2016) **281**:28–38. doi: 10.1016/j.amc.2016.01.049
50. Khalid N, Abbas M, Iqbal MK, Baleanu D. A numerical investigation of Caputo time fractional Allen-Cahn equation using redefined cubic B-spline functions. *Adv Differ Equat*. (2020) **2020**:1–22. doi: 10.1186/s13662-020-02616-x
51. De Boor C. On the convergence of odd-degree spline interpolation. *J Approx Theory*. (1968) **1**:452–63. doi: 10.1016/0021-9045(68)90033-6
52. Hall CA. On error bounds for spline interpolation. *J Approx Theory*. (1968) **1**:209–18. doi: 10.1016/0021-9045(68)90025-7
53. Iqbal MK, Abbas M, Zafar B. New quartic B-spline approximations for numerical solution of fourth order singular boundary value problems. *Punjab Univ J Math*. (2020) **52**:47–63.
54. Wasim I, Abbas M, Amin M. Hybrid B-spline collocation method for solving the generalized Burgers-Fisher and Burgers-Huxley equations. *Math Probl Eng*. (2018) **2018**:6143934. doi: 10.1155/2018/6143934

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Amin, Abbas, Iqbal and Baleanu. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.