Kashf Journal of Multidisciplinary Research
Vol: 02 - Issue 09 (2025)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

NUMERICAL APPROXIMATION OF TIME-FRACTIONAL KLEIN–GORDON EQUATION USING B-SPLINE COLLOCATION TECHNIQUES

Javeria Kousar*
Faculty of Sciences, The Superior University, Lahore, Pakistan

Muhammad Amin
Faculty of Sciences, The Superior University, Lahore, Pakistan

Manzar Abbas
Faculty of Sciences, The Superior University, Lahore, Pakistan

Ayesha Samra
Faculty of Sciences, The Superior University, Lahore, Pakistan

[*]***Corresponding Author:*** iamjaveria717@gmail.com

Article Info

Abstract

This study proposes a numerical technique based on a hybrid cubic B-Spline functions for obtaining approximate solutions to the time-fractional Klein-Gordon equation. Fractional derivative is discretized using a finite difference approach in Caputo sense, while the spatial domain is handled through the application of a hybrid cubic B-Spline scheme on a structured grid. To assess the accuracy and performance of the method, some computational experiments have been conducted. The outcomes demonstrate that the proposed technique delivers superior accuracy and computational efficiency when compared to several existing methods.

Keywords:

Time-fractional Klein-Gordon equation, Caputo derivative, Hybrid cubic B-Spline, Finite difference method, Fractional differential equations, Collocation technique.

Introduction

Fractional-order differential equations have emerged as a significant area of study because of their wide-ranging applications in modeling complex systems. They are widely employed in disciplines such as traffic flow analysis, earthquake prediction, physical process simulation, signal processing, financial modeling, control systems, fractional dynamics, and various forms of mathematical modeling[1-6]. Unlike classical integer-order models, fractional-order formulations are capable of incorporating memory and hereditary effects, making them highly effective for describing real-world phenomena with long-term dependencies.

Over the past few decades, research on fractional-order differential equations has expanded rapidly, leading to the development of a variety of analytical and numerical techniques to address such problems[7-19]. This progress has been driven by the necessity to obtain accurate and efficient solutions for models that cannot be solved using conventional approaches. The diversity of problems in science and engineering has encouraged the creation of specialized methods, each offering distinct advantages depending on the nature of the application.

Several definitions of fractional derivatives have been introduced, each suited to different theoretical frameworks and physical interpretations. Comprehensive discussions and comparative analyses of these definitions can be found in the literature[20-26].

In this study, we focus on the time-fractional nonlinear Klein–Gordon equation (KGE), which plays an important role in describing nonlinear wave propagation and dynamic behaviors in various physical systemsplays a pivotal role in modeling relativistic wave phenomena and scalar field dynamics, particularly within the framework of quantum mechanics and other areas of theoretical physics. In recent years, its generalized form where the classical time derivative is replaced by a fractional-order derivative has garnered considerable interest. This time-fractional Klein-Gordon equation (TFKGE) offers a more versatile and accurate framework for representing complex physical processes that exhibit memory-dependent and hereditary behavior, such as those encountered in viscoelastic materials, anomalous diffusion, and systems characterized by nonlocal interactions.

Analytical solutions of the TFKGE are often challenging to obtain due to the inherent no locality of fractional derivatives. As a result, numerical methods have become increasingly essential for approximating solutions to such equations. Among various numerical strategies, spline-based techniques particularly those using cubic splineshave demonstrated significant promise. These methods benefit from the smoothness and continuity of cubic spline basis functions, which provide stable and accurate approximations, especially for differential equations involving higher-order spatial terms.

In this study, we explore a numerical approach for solving the time-fractional Klein–Gordon equation by employing a collocation method based on cubic spline functions. The fractional derivative is interpreted in the Caputo sense, which is well-suited for physical applications due to its compatibility with standard initial conditions. The time discretization is achieved using finite difference schemes, while the spatial discretization is handled using cubic spline basis functions.

The application of fractional differential equations spans a wide range of scientific disciplines, including transport processes, geophysics, signal processing and financial modeling, engineering systems, and applied mathematics. Their relevance stems from their ability to accurately capture complex behaviors not easily described by classical models. This growing interest has led to an expanding body of research focused on both analytical and numerical treatment of such equations. Numerous techniques have been proposed to handle the intricacies of fractional calculus, each grounded in different definitions of fractional derivatives, such as the Caputo, Riemann–Liouville, and Grünwald–Letnikov forms.

In this work, we focus on the time-fractional nonlinear Klein–Gordon equation expressed in the general form:

$$\frac{\partial^{\beta}}{\partial t^{\beta}}u(y,t)+\rho\frac{\partial^{2}}{\partial y^{2}}u(y,t)+\rho_1 u(y,t)+\rho_2 u^{\sigma}(y,t)=f(y,t)\quad 0<y\le M,\ t_0<t\le T \tag{1}$$

$$u(y,t_0)=\phi_1(y),\quad u_t(y,t_0)=\phi_2(y) \tag{2}$$

$$u(0,t)=\phi_3(t),\quad u(M,t)=\phi_4(t) \tag{3}$$

WhereCaputo fractional time derivative is represented by$\frac{\partial^{\beta}}{\partial t^{\beta}}$and $u= u(y,t)$represents the displacement of the wave at (y,t). $\beta\in(1,2]$show the fractional order of the time derivative while $f(y,t)$represents to source term. Here ρ,ρ_1 and ρ_2 are real numbers, and sigma is equal to $\sigma=2$ or 3.

The time-fractional Klein–Gordon equation (TFKGE) has been utilized in various fields such as condensed matter physics, quantum field theory, and nonlinear dynamics. Multiple analytical and semi-analytical techniques have been proposed for solving such equations, including the Adomian decomposition method, variational iteration methods, homotopy analysis methods[27-30] and spectral approaches. For instance, Jafari et al. demonstrated the use of fractional B-Spline functions for constructing semi-analytical solutions[31]. Vong and Wang[32, 33] developed compact finite difference schemes for both one and two-dimensional TFKGE problems, confirming their stability and convergence through energy-based analyses[34]. In another direction, Dehghan introduced a meshless method using extended basis functions in fully implicit schemes to solve fractional Klein–Gordon and sine-Gordon equations[35].

Further developments include the application of iterative correction procedures and the Adomian decomposition technique by Jafari to obtain approximate solutions to time-fractional KGEs[36].Chanet al. addressed nonlinear fractional differential equations using spectral methods[37], while Lyu and Vong proposed a linearized second-order approach specifically designed for fractional nonlinear Klein–Gordon equations[38]. Nagy [39]introduced a sinc-Chebyshev collocation method, which merges sinc functions with second order shifted Chebyshev polynomials to effectively tackle nonlinear TFKGEs. Sahu and Jena employed a hybrid scheme combining a Newton–Raphson method with a modified Laplace-Adomian decomposition technique for solution approximation[40].Yaseen et al. proposed a trigonometric B-spline collocation scheme to enhance the accuracy of nonlinear TFKGE solutions[41, 42]. More recently, Vivas-Cortez etal. improved cubic spline-based methods by integrating an extended spline formulation with the Crank–Nicolson scheme, establishing numerical stability and convergence[43].

In this work, we introduce a numerical method to solve the time-fractional Klein–Gordon equation using newly developed restructured Hybrid Cubic B-Spline (HCBS) basis functions. These functions represent a simplified version of traditional cubic B-splines, with an additional adjustable parameter that enhances adaptability to the solution's shape. The Caputo fractional derivative is discretized using a central difference scheme, while spatial interpolation is achieved through HCBS basis functions.

The paper is structured as follows: Section II presents the mathematical formulation of the Caputo fractional derivative and the time discretization using finite differences. The subsequent sections provide implementation details of the HCBS method, its stability analysis, numerical results, and a comparison with existing techniques.

1. Time Discretization

Let consider the time domain [0, T] which is equally divide into Q subintervals of length $\Delta t = \frac{T}{Q}$ .Here, endpoint is $0 = t_0 < t_1 < \dots,\ t_Q = T$ and $t_q = q\Delta t$ and $q = 0:1:Q$. Firstly, Caputo fractional derivative is discretized at $t = t_{q+1}$ [44].

$$\frac{\partial^{\beta} u(y,t_{q+1})}{\partial t^{\beta}} = \frac{1}{\Gamma(2-\beta)} \int_0^{t_k} \frac{\partial^2 u(y,p)}{\partial p^2} (t_{q+1}-p)^{-\beta+1} dp \ , \qquad (1<\beta\le 2)$$

$$= \frac{1}{\Gamma(2-\beta)} \sum_{k=0}^{r} \int_{t_k}^{t_{k+1}} \frac{\partial^2 u(y,p)}{\partial p^2} (t_{q+1}-p)^{-\beta+1} dp$$

$$= \frac{1}{\Gamma(2-\beta)} \sum_{k=0}^{r} \frac{u(y,t_{k+1})-2u(y,t_k)+u(y,t_{k-1})}{\Delta u^2} \int_{t_k}^{t_{k+1}} (t_{q+1}-p)^{-\beta+1} dp + J_{\Delta t}^{q+1} \tag{4}$$

$$= \frac{1}{\Gamma(2-\beta)} \sum_{k=0}^{q} \frac{u(y,t_{k+1})-2u(y,t_k)+u(y,t_{k-1})}{\Delta t^2} \int_{t_{q-k}}^{t_{q-k+1}} (\in)^{-\beta+1} d \in + J_{\Delta t}^{q+1}$$

$$= \frac{1}{\Gamma(3-\beta)} \sum_{k=0}^{q} \frac{u(y,t_{q-k+1})-2u(y,t_{q-k})+u(y,t_{q-k-1})}{\Delta t^{\beta}} (k+1)^{2-\beta} - k^{2-\beta} + J_{\Delta t}^{q+1}$$

$$= \frac{1}{\Gamma(3-\beta)} \sum_{k=0}^{q} a_k \frac{u(y,t_{q-k+1})-2u(y,t_{q-k})+u(y,t_{q-k-1})}{\Delta t^{\beta}} + J_{\Delta t}^{q+1}$$

Where $a_k = (k+1)^{2-\beta} - k^{2-\beta}, \sigma = (t_{r+1} - a)$ and $J_{\Delta t}^{q+1}$. The truncation error is bound i.e.

$$\left| J_{\Delta t}^{q+1} \right| \le \varphi(\Delta t)^{2-\beta} \tag{5}$$

Where φ Is constant the coefficients a_k which are,

- The a_k't are non-negative for $(K = 0, 1,\ 2\dots,\ R)$.
- $1 = A_0 > A_1 > A_2 \dots\dots A_n$ and $A_n \to 0$ as $n \to \infty$.
- $(2a_0 - a_1) + \sum_{k=1}^{q-1} \ (-A_{k+1}\ + 2A_k - A_{k-1}) + (2A_q - Aq - 1) - q_r = 1$.

Putting equation (4) into (1)

$$\frac{1}{\Gamma(3-\beta)(\Delta t)^{\beta}} \sum_{k=0}^{q} A_k (y,t_{q-k+1}) - 2u(y,t_{q-k}) + u(y,t_{q-k-1}) + \rho u_{yy}(y,t) + \rho_1 u(y,t) + \rho_2 u^{\sigma}(y,t) = f(y,t)$$

$$q = 0,\ 1,\ 2,\ 3, \dots, Q-1 \tag{6}$$

Suppose $\alpha = \frac{1}{\Gamma(3-\beta)\Delta t^{\beta}}$ and $u(y,t_{q+1}) = u^{q+1}$

Applying a θ weighted equation (6) takes from

$$\beta\rho_0(u^{q+1} - 2u^q + u^{q-1}) + \beta\sum_{k=1}^{q}\rho_k(u^{q-k+1} - 2u^{q-k} + u^{q-k-1}) + \vartheta(\rho u_{yy}^{q+1} + \rho_1 u^{q+1}) = f^{q+1} - (1-\vartheta)(\rho u^q{}_{yy} + \rho_1 u^q) - \rho_2(u^{\sigma})^q$$

$$q = 0,\ 1,\ 2,\ldots,Q-1 \tag{7}$$

For θ =1 we get semi directional numerical system

$$(\beta\rho_0 + \rho_1)u^{q+1} + \rho u_{yy}^{q+1} = 2\beta p_0 v^q + \beta\sum_{k=1}^{q}\rho_k(u^{q-k+1} - 2u^{q-k} + u^{q-k-1}) - \rho_2(u^{\sigma})^q - \beta\rho_0 v^{q-1} + f^{q+1}$$

$$q = 0,\ 1,\ 2,\ldots,Q-1 \tag{8}$$

2. Extended cubic B-spline functions

Consider the spatial interval $[c,\ d]$ divided into N equal segments, each of length $h = (d-c)/N$ this result in boundary points $c = c_o < c_1 \ldots < c_n = d$, where each point is defined as $c_n = c_0 + ch$ for $n = 0,\ 1,\ldots,N$ for a function $U(y,\ t)$ that is sufficiently continuous, there is always a unique extension available[45]

$$u^*(y,t) = \sum_{n=-1}^{n+1}\zeta_n(t)t_n(y,\lambda) \tag{9}$$

$$T_n(y,\lambda) = \frac{1}{24h^4}\begin{cases} 4h(y-y_{n-2})^3(1-\lambda) + 3(y-y_{n-2})^4\lambda, & y \in [y_{n-2}, y_{n-1}) \\ h^4(4-\lambda) + 12h^3(y-y_{n-1}) + 6h^2(y-y_{n-1})^2(2+\lambda) & \\ \quad -12h(y-y_{n-1})^3 - 3(y-y_{n-1})^4\lambda, & y \in [y_{n-1}, y_n) \\ h^4(4-\lambda) - 12h^3(y-y_{n+1}) - 6h^2(y-y_{n+1})^2(2+\lambda) & \\ \quad +12h(y-y_{n+1})^3 + 3(y-y_{n+1})^4\lambda, & y \in [y_n, y_{n+1}) \\ -4h(y-y_{n+2})^3(1-\lambda) - 3(y-y_{n+2})^4\lambda, & y \in [y_{n+1}, y_{n+2}) \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Here λ with $-n\ (n-2) \le \lambda \le 1$ is a real number responsible for fine tuning the curve and n gives the degree of ECBS used to generate different forms of ECBS functions the approximate solution $(U^*)_n^q = U^*(y_n, t^q)$ and its first two derivatives can be expressed with respect to the spatial variable y at the q^th^time step as follow[46],

$$
\begin{aligned}
(u^*)_n^q &= a_1\varsigma_{n-1}^q + a_2\varsigma_n^q + a_1\varsigma_{n+1}^q \\
(u_y)_n^q &= a_3\varsigma_{n-2}^q - a_3\varsigma_{n+1}^q \\
(u_{yy}^*)_n^q &= a_4\varsigma_{n-1}^q + a_5\varsigma_n^q + a_4\varsigma_{n+1}^q
\end{aligned} \tag{11}
$$

where, $a_1 = \frac{4-\lambda}{24}$, $a_2 = \frac{16+2\lambda}{24}$, $a_3 = \frac{-1}{2h}$, $a_4 = \frac{2+\lambda}{2h^2}$ and $a_5 = \frac{-4-2\lambda}{2h^2}$.

3. Redefined extended cubic B-Spline function

The basic functions $t_{-1}, t_0, \ldots, t_{M+1}$ in a standard ECBS collocation method, do not satisfy the boundary conditioned when Dirichlet type end conditioned are applied ,as they do not vanish at the spatial domain boundaries to address this issue, we need to modify these basis functions to ensure that they do vanish at the boundaries this is achieved by introducing a weight function $\phi(y,s)$ which effectively removes $\varphi(y,t) = \frac{t_{-1}(y,\eta)}{t{-}1(y_0,\eta)}\phi_3(t) + \frac{t_{N+1}(y,\eta)}{T_{N+1}(y_N,\eta)}\phi_4(t)$ ς_{-1} a ζ_{n+1} from equation (9) as[47]:

$$
u(y,t) = \varphi(y,t) + \sum_{n=0}^{N} \zeta_n(t) t_n(y,\lambda) \tag{12}
$$

Where the weight function (y,t) and the redefined ECBS (RECBS) functions are given by

$$
\varphi(y,t) = \frac{t_{-1}(y,\lambda)}{t_{-1}(y_0,\lambda)}\phi_3(t) + \frac{t_{N+1}(y,\lambda)}{T_{N+1}(y_N,\lambda)}\phi_4(t) \tag{13}
$$

$$
\left\{\begin{array}{ll}
\left\{ t_n(y,\lambda) = t_n(y,\lambda) - \dfrac{t_n(y_0,\lambda)}{t_{-1}(y_0,\lambda)} t_{-1}(y,\lambda) \right. & \text{for } n = 0,1 \\
\left\{ t_n(y,\lambda) = t_n(y,\lambda) \right. & \text{for } n = 2{:}1{:}N\text{-}2, \\
\left\{ t_n(y,\lambda) = t_n(y,\lambda) - \dfrac{t_n(y_n,\lambda)}{T_{N+1}(y_N,\lambda)} T_{N+1}(y,\lambda) \right. & \text{for } n = N\text{-}1, N.
\end{array}\right\} \tag{14}
$$

4. Space Discretization

Putting equation (12) in equation (8) at $t = t_{q+1}$ we get,

$$
(\alpha w_0 + w_1)u^{q+1} + w u_{yy}^{q+1} = 2\alpha w_0 U^q + \alpha \sum_{K=1}^{q} w_K (U^{q-k+1} - 2U^{q-K} + U^{q-k-1}) - W_2 (U^\sigma)^q - \alpha w_0 U^{q-k} + f^{q+1} \tag{15}
$$

Discretizing at $y = y_i$ we get

$$
(\alpha + w_1)U_i^{q+1} + w\left(U_{xx}\right)_i^{q+1} = 2\alpha U_i^q + \alpha \sum_{K=1}^{q} w_k (U_i^{q-k+1} - 2U_i^{q-K} + U_i^{q-k-1}) - w_2 (U^\sigma)_i^q - \alpha U_i^{q-1} + f_i^{q+1},
$$

$$
(i = 0,\ 1, 2, \ldots, N) \tag{16}
$$

Using (12) the last expression takes the form

$$(\alpha + w_1)\left[\varphi_i^{q+1} + \sum_{n=0}^{N} \zeta_n^{q+1} t_n(y_i,\lambda)\right] + \partial\left[(\varphi_{yy})_i^{q+1} + \sum_{n=0}^{N} \xi_n^{q+1} t_n(y_i,\lambda)\right]$$

$$= 2\alpha u_i^q + \alpha \sum_{k=1}^{q} w_k (u_i^{q-k+1} - 2u_i^{q-k} + u_i^{q-k-1}) - w_2 (u^\sigma)_i^q - \alpha u_i^{q-1} + f_i^{q+1}, \quad \mathrm{i} = 0,\ 1,\ 2,\ldots, N \tag{17}$$

So, we obtain the following structure of $N+1$ equation in $N+1$ unknown.

$$\begin{pmatrix} b_1^* & & & & & & & \\ b_1 & b_2 & b_1 & & & & & \\ & b_1 & b_2 & b_1 & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & & b_1 & b_2 & b_1 & \\ & & & & & b_1 & b_2 & b_1 \\ & & & & & & & b_1^* \end{pmatrix} \begin{pmatrix} \zeta_0^{q+1} \\ \zeta_1^{q+1} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \zeta_{N-1}^{q+1} \\ \zeta_N^{Q+1} \end{pmatrix} = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ x_{N-1} \\ x_N \end{pmatrix} \tag{18}$$

Where

$$b_1^* = \frac{12w(\lambda+2)}{h^2(\lambda-4)}, \quad b_1 = \frac{h^2(\alpha+w_1)(\lambda-4)+12w(\lambda+2)}{24h^2}$$

$$b_2 = \frac{h^2(\alpha+w_1)(\lambda+8)-12w(\lambda+2)}{12h^2}$$

$$x_i = 2\alpha u_i^q + \alpha \sum_{k=1}^{q} w_k (U_i^{q-k+1} - 2U_i^{q-k} + U_i^{q-k-1} - w_2 (U^\sigma)_i^q - \alpha U_i^{q-1} + \psi_i^{q+1}$$

$$\psi_i^q = f_i^q - (\alpha + w_1)\varphi_i^q - w(\varphi_{yy})_i^q$$

For numerical procedure, apply the given initial conditioned to acquire the set of equations.

$$\begin{cases} (U')_n^o = \varphi'(x_n) & for\, n = 0, \\ (U)_n^o = \varphi(x_n) & for\, n = 1:1:M-1, \\ (U')_n^o = \varphi'(x_n) & for\, n = N. \end{cases} \tag{19}$$

The matrix representation of (19) is

$$\begin{pmatrix} a_1^* & a_2^* & & & & & \\ a_1 & a_2 & a_1 & & & & \\ & a_1 & a_2 & a_1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & a_1 & a_2 & a_1 & \\ & & & & a_1 & a_2 & a_1 \\ & & & & & -a_2^* & -a_1^* \end{pmatrix} \begin{pmatrix} \zeta_0^0 \\ \zeta_1^0 \\ \vdots \\ \vdots \\ \vdots \\ \zeta_{N-1}^0 \\ \zeta_N^0 \end{pmatrix} = \begin{pmatrix} (\phi_1^{'})_0 - (\phi^{'})_0^0 \\ (\phi_1)_1 - \phi_1^0 \\ \vdots \\ \vdots \\ \vdots \\ (\phi_1)_{N-1} - \phi_{N-1}^0 \\ (\phi_1^{'})_N - (\phi^{'})_N^0 \end{pmatrix} \tag{20}$$

Where $a_1^* = \frac{8+\lambda}{h(4-\lambda)}$, $a_2^* = \frac{1}{h}$..We solve (20) to obtain $\left[\zeta_0^0, \zeta_1^0, \ldots\ldots \zeta_N^0\right]^S$.Then put the values of ζ_i into equation (12) to find out the value of Y^0.We can use equation (18) for $q = 0, 1, 2, \ldots\ldots Q-1$, but the term involving y^{-1} appears in equation (18) for $q = 0$. To resolve this issue by using the velocity condition given in (2).

$$Y^{-1} = Y^0 - \Delta t \varphi_2(y).$$

5. Stability Analysis

Fourier method is used to check the stability of the proposed numerical method. Let ε_m^q and $\tilde{\varepsilon}_m^q$

Exact and approximate growth factors of the Fourier modes ω_n^q and $\tilde{\omega}_n^q$ respectively. The correct error e_n^q is given by

$$e_n^q = \varepsilon_n^q - \tilde{\varepsilon}_n^q \qquad n = 1:1:M-1, \qquad q = 0:1:Q \tag{21}$$

Where $e^q = \left[e_1^q, e_2^q, \ldots\ldots e_{N-1}^q\right]^s$

$$\begin{aligned}
&(\alpha a_1 + w_1 a_1 + w a_4) e_{n-1}^{q+1} + (\alpha a_2 + w_1 a_2 + w a_5) e_n^{q+1} + (\alpha a_1 + w_1 a_1 + w b_4) e_{n+1}^{q+1} \\
&= 2\alpha(a_1 e_{n-1}^q + a_2 e_n^q + a_1 e_{n+1}^q) - \alpha(a_1 e_{n-1}^{q-1} + a_2 e_n^{q-1} + a_1 e_{n+1}^{q-1}) - \alpha \sum_{k=1}^{q} w_k [a_1(e_{n-1}^{q-k+1} - 2e_{n-1}^{q-k} + e_{n-1}^{q-k-1}) \\
&+ a_2(e_n^{q-k+1} - 2e_n^{q-k} + e_{n-1}^{q-k-1}) + a_1(e_n^{q-k+1} - 2e_{n+1}^{q-k} + e_{n+1}^{q-k-1})
\end{aligned} \tag{22}$$

The error equations satisfied the end conditions

$$e_n^0 = \phi_1(y_n), \qquad n = 1:1:N, \tag{23}$$

And

$$e_0^q = \phi_3(t_q), e_n^q = \phi_4(t_q), \qquad q = 0:1:Q \tag{24}$$

We define the grid function as

$$e^q = \begin{cases} e_n^q & \text{if } y_n - \frac{g}{2} < y \leq y_n + \frac{g}{2}, \text{ for } n = 1:1:N-1, \\ 0 & \text{if } b \leq y \leq \frac{2b+g}{2} \text{ or } \frac{2a-g}{2} \leq y \leq a. \end{cases} \tag{25}$$

Now

$$e^q(y) = \sum_{q=-\infty}^{\infty} e_q(m) \varepsilon^{\frac{2\pi i n y}{a-b}}, q = 1:1:Q \tag{26}$$

Where,

$$\varepsilon_q(m)=\frac{1}{a-b}\int_b^a e^q(y)\varepsilon^{\frac{-2\pi iny}{a-b}}dx \tag{27}$$

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

$$\left\|e^q\right\|_2=\Big(\sum_{m=1}^{Q-1} g\left|e_m^q\right|^2\Big)^{\frac{1}{2}}=\Big(\int_b^{b+\frac{g}{2}}\left|e^q\right|^2 dx+\sum_{m=1}^{Q-1}\int_{y_m-\frac{h}{2}}^{y_m+\frac{h}{2}}\left|e^q\right|^2 dx+\int_{a-\frac{g}{2}}^{a}\left|e^q\right|^2 dx\Big)^{\frac{1}{2}}$$

$$=\Big(\int_b^a\left|e^q\right|^2 dx\Big)^{\frac{1}{2}}$$

From Perceval equality we have $\int_b^a\left|e^q(m)^2\right|dx=\sum_{-\infty}^{\infty}\left|\varepsilon_m(n)\right|^2$ so the above expression can be written as

$$\left\|e^q\right\|_2^2=\sum_{q=-\infty}^{\infty}\left|e_q(m)\right|^2 \tag{28}$$

Now, the solution of Fourier series,

$$e_k^q=\varepsilon_q\varepsilon^{ivkg} \tag{29}$$

Where, $i=\sqrt{-1}$ And $u=\frac{2\pi v}{a-b}$ using equation (29) in equation (22) and then dividing by e^{iukg}

$$\begin{aligned}
&(\alpha a_1+w_1a_1+wa_4)\varepsilon_{q+1}e^{-iug}+(\alpha a_2+w_1a_2+wa_5)\varepsilon_{q+1}+(\alpha a_1+w_1a_1+wa_4)\varepsilon_{q+1}e^{iug}\\
&=2\alpha(a_1\varepsilon_q e^{-iug}+a_2\varepsilon_q+a_1\varepsilon_q e^{-iug})-\alpha(a_1\varepsilon_{q-1}e^{-iuh}+a_2\varepsilon_{q-1}+a_1\varepsilon_{q-1}e^{iug})+a_2(\varepsilon_{q-k+1}e^{-iug}-2\varepsilon_{q-k}e^{iug}+\varepsilon_{q-k-1}e^{iug})\\
&-\alpha\sum_{k=1}^{q}w_k[a_1(\varepsilon_{q-k+1}e^{-iug}-2\varepsilon_{q-k}+\varepsilon_{q-k-1}e^{iug})+a_2(\varepsilon_{q-k+1}-2\varepsilon_{q-k}+\varepsilon_{q-k-1})+a_1(\varepsilon_{q-k+1}e^{iug}-2\varepsilon_{q-k}e^{iug}+\varepsilon_{q-k-1}e^{iug})
\end{aligned} \tag{30}$$

After collecting like terms, we know that $e^{iug}+e^{-iug}=2\cos(ug)$. So, we get.

$$\varepsilon_{q+1}=\frac{1}{\lambda}[2\varepsilon_q-\varepsilon_{q-1}-\sum_{k=1}^{q}\mathrm{P}_k\left(\varepsilon_{q-k+1}\right)(a_1+a_2)\ \varepsilon_{q-k}+\varepsilon_{q-k-1}] \tag{31}$$

Where,

$$\lambda=1+\frac{\mathrm{P}_1}{\alpha}+\frac{12\mathrm{P}(2+u)\sin^2(^{ug}/_2)}{\alpha g^2\{-6+(4-u)\sin^2(^{ug}/_2)\}} \qquad \lambda\geq 1 \text{ for u>-2}$$

6. Convergence of the Scheme

We examine the merging of CIMSS by doing the following. The strategy Khalid et al. [48]depict in their ponder. At this point, wear starting using the taking after valuable hypotheses[49, 50].

Theorem 1. Suppose, $\eta=\{a=x_0,x_1,......,x_M=b\}$ which refer to a partition and we have $[a,b]$ for $m=0,1,2,......,M$ and $v\in C$ with $x_M=mh$. Suppose, $v\in C^4[a,b]$ and $f\in C^2[a,b]$. Suppose, $V(x,t)$ is the spline that presents the preparation bend of this issue. At that point there exist constants which are not dependent on h. so,

$$\left\|\xi^i(u(y,t)-U(y,t)\right\|_\infty \le E_i g^{4-i}\ \forall t\ge 0,\qquad i=0,1,2 \tag{32}$$

Lemma: The extended B-spline gratify the difference $\sum_{n=0}^{N}|t_n(y,\eta)|\le 1.75$ for $0\le y\le 1$ (33)

Proof: By the triangle inequality we have

$$\left|\sum_{n=0}^{N}t_n(y,\eta)\right|\le\sum_{n=0}^{N}|t_n(y,\eta)|$$

For any knot y_n ,we have

$$\sum_{n=0}^{N}|t_n(y,\eta)|=|t_{n-1}(y_n,\eta)|+|t_n(y_n,\eta)|+|t_{n+1}(y_n,\eta|=1<\tfrac{7}{4}$$

We obtain

$$t_n(y_n,\eta)=\tfrac{1}{12}(8+\eta),t_{n-1}(y_{n-1,}\eta)=\tfrac{1}{12}(8+\eta)$$
$$t_{n+1}(y_n,\eta)=\tfrac{1}{24}(4-\eta),t_{n-2}(y_{n-1},\eta)=\tfrac{1}{24}(4-\eta)$$

Then for $y\in|y_{n-1},y_n|,t_n(y,\eta),t_{n-1}(y,\eta)$ area bounded above by $\frac{1}{12}(8+\eta)$

Similarly, $t_{n+1}(y,\eta),t_{n-2}(y,\eta)$ $t_{n+1}(y,\eta),t_{n-2}(y,\eta)$ are bounded above by $\frac{1}{24}(4-\eta)$

for any point $y_{n-1}\le y\le y_n$ $y_{n-1}\le y\le y_n$ we obtain

$$\sum_{n=0}^{N}|t_n(y,\eta)|=|t_{n-1}(y,\eta)|+|t_n(y,\eta)+|t_{n+1}(y,\eta)||+|t_{n-2}(y,\eta)|=\tfrac{1}{12}(\eta+20)$$

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

$$|t_n(y,\eta)|\le 1.75$$

Theorem 2: The extended cubic B-splineestimate$u(y,t)$ the analytical exact solution $u(y,t)$and if $E\in c^2[0,1]$ then $\|u(y,t)-u(y,t)\|_\infty\le E g^2\forall t\ge 0$

$$E\ \|u(y,t)-u(y,t)\|_\infty\le E g^2\forall t\ge 0 \tag{34}$$

Where g is reasonably small and E >0 is a constant not depending on g

Proof: Suppose, $u(y,t)=\sum_{n=0}^{N} dn(t)\lambda_n(y)$ is a spline which is calculated for the estimated solution of $U(y,t)$ and the exact solution $u(y,s)$. Suppose, if $lu(y_n,t)=y^{\square}(y_n,t)$ for $n=0:1:N$ be the collocation c conditions then $lu^{\square}(y,t)=y^{\square}(y_n,t), n=0:1:N$. Now, the problem can be explained in the form of a difference equation $L(U(y_n,t)-U(y_n,t))$ as,

$$(\alpha a_1+w_1a_1+wa_4)\varsigma_{n-1}^{q+1}+(\alpha a_2+w_1a_2+wa_5)\varsigma_n^{q+1}+(\alpha a_1+w_1a_1+wa_4)\varsigma_{n+1}^{q+1} \quad (35)$$

$$=2\alpha(a_1\varsigma_{n-1}^{q}+a_2\varsigma_n^{q}+a_1\varsigma_{n+1}^{q})-\alpha(a_1\varsigma_{n-1}^{q-1}+a_2\varsigma_n^{q-1}+a_1\varsigma_{n+1}^{q-1})-\alpha\sum_{k=1}^{q}p_k[a_1(\varsigma_{n-1}^{q-k+1}-2\varsigma_{n-1}^{q-k}+\varsigma_{n-1}^{q-k-1})$$
$$+a_2(\varsigma_n^{q-k+1}-2\varsigma_n^{q-k}+\varsigma_n^{q-k-1})+a_1(\varsigma_{n+1}^{q-k+1}-2\varsigma_{n+1}^{q-k}+\varsigma_{n+1}^{q-k-1})]+\tfrac{1}{g^2}\lambda_n^{q+1}$$

The boundary conditions can be written as $a_1\varsigma_{n-1}^{q+1}+a_2\varsigma_n^{q+1}+a_1\varsigma_{n+1}^{q+1}=0, n=0,N,$

Where $\varsigma_n^q=\varsigma_n^q-d_n^q, n=0:1:N$ $\lambda_n^q=g^2[x_n^q-x^{\square}{}_n^q], n=0:1:N$ $\varsigma_n^q=\varsigma_n^q-d_n^q, n=0:1:N$

And $\lambda_n^q=g^2[x_n^q-x^{\square}{}_n^q], n=0:1:N$.

We have

$$\left|\lambda_n^q\right|=g^2\left|x_n^q-x^{\square}{}_n^q\right|\le Eg^4 \quad \left|\lambda_n^q\right|=g^2\left|x_n^q-x^{\square}{}_n^q\right|\le Eg^4$$

We describe $\lambda^q=\max\{\left|\lambda_n^q\right|:0\le m\le M\}, e^{\square}{}_n^q=\left|\xi_n^q\right|, e^{\square q}=\max\{\left|e_n^q\right|:0\le m\le M\}$ for $q=0$ equation converts into the succeeding relation.

$$(\alpha a_1+wa_1+wa_4)\varsigma_{n-1}^{1}+(\alpha a_2+w_1a_2+wa_5)\varsigma_n^{1}+(\alpha a_1+w_1a_1+wa_4)\varsigma_{n+1}^{1}$$

$$=(\alpha+w_1)(a_1\varsigma_{n-1}^{0}+a_2\varsigma_n^{0}+a_1\varsigma_{n+1}^{0})+\tfrac{1}{g^2}\lambda_n^1$$

We get by using initial condition $e^0=0$,

$$(\alpha a_2+w_1a_2+wa_5)\varsigma_n^1=(\alpha a_1+wa_4)(\varsigma_{n+1}^1-\varsigma_{n-1}^1)+w_1a_1(\varsigma_{n+1}^1-\varsigma_{n-1}^1)+\tfrac{1}{g^2}\lambda_n^1$$

By putting absolute values of $\lambda_n^q, \varsigma_n^1$ and small g. then we get,

$$e^{\square}{}_n^1\le\frac{6Eg^4}{\alpha g^2(\eta+2)+12(-2-\eta)+p_1g^2(2+\eta)}$$

By using boundary conditions, we accomplish that

$$e^{\square 1}\le E_1g^2$$

Where E_1 is independent of the spatial method, we suppose that $e_n^{k} \leq E_k g^2$ true for $k=1:1:q$. let $E = \max\{E_k : 0 \leq K \leq q\}$ then

$$(\alpha a_1 + pa_1 + wa_4)\varsigma_{n-1}^1 + (\alpha a_2 + w_1 a_2 + wa_5)\varsigma_n^1 + (\alpha a_1 + w_1 a_1 + wa_4)\varsigma_{n+1}^1$$

$$+(\alpha a_1 + w_1 a_1 + wa_4)\varsigma_{n-1}^{q+1} + (\alpha a_2 + w_1 a_2 + wa_5)\varsigma_n^{q+1} + (\alpha a_1 + w_1 a_1 +$$

$$(w_1 - 2w_2 + w_3)(a_1\varsigma_{n-1}^{q-1} + a_2\varsigma_n^{q-1} + a_1\varsigma_{n-1}^{q-1}) + ... + (w_{q-4} - 2w_{q-3} + w_{q-2})(a_1\varsigma_{n-1}^1 + a_2\varsigma_n^1 + a_1\varsigma_{n+1}^1)$$
$$+w_{q-1}(a_1\varsigma_{n-1}^0 + a_2\varsigma_n^0 + a_1\varsigma_{n+1}^0)] + \tfrac{1}{g^2}\lambda_n^{q+1}$$

again, taking absolute values of η_n^q, ξ_{n+1}^0 we have

$$e_n^{1} \leq \frac{6Eg^4}{\alpha g^2(\eta+2) + 12(-2-\eta) + p_1 g^2(2+\eta)}$$

$$[2\alpha(a_1\varsigma_{n-1}^q + a_2\varsigma_n^q + a_1\varsigma_n^q) - \alpha\sum_{k=0}^{q-1}(w_{k+1} - 2w_k - w_{k-1})Eg^2 + Eg^2]$$

By using the boundary conditions, we get $e_n^{q+1} \leq Eg^2$ $e_n^{q+1} \leq Eg^2$

Hence, for all values of n,

$$e_n^{q+1} \leq Eg^2 \tag{36}$$

Now, $u(y,t) - u(y,t) = \sum_{n=0}^{N}(d_n(t) - \varsigma_n(t)t_n(y)$ s

Taking the infinity norm and applying lemma we obtain

$$\left\|u(y,t) - u(y,t)\right\|_\infty \leq 1.75Eg^2 \tag{37}$$

Making use of the triangle inequality, we get

$$\|u(y,t) - u(y,t)\|_\infty \leq \quad \|u(y,t) - u(y,t)\|_\infty + \|u(y,t) - u(y,t)\|_\infty \tag{38}$$

Using the inequalities 32) and (37) in (38) we obtain

$$\|u(y,t) - u(y,t)\|_\infty \leq E_0 g^4 + 1.75Eg^2 = E g^2$$

Where, $E = E_0 g^2 + 1.75E$

$\lambda(\Delta t)^{2-\beta}$

We accomplish that the numerical approach converges unconditionally, therefore

$$\|u(y,t) - u(y,t)\|_\infty \leq Eg^2 + \lambda(\Delta t)^{2-\beta}$$

Where $E^{\square}$ is a constant and $\beta \in (1,2]$ hence, hypothetically the projected system is $o(g^2 + \Delta t^{2-\beta})$ accurate.

7. NumericalResults and Discussion.

To evaluate the accurateness and performance of the projected numerical scheme, several benchmark test problems are considered. The errors are measured using both the maximum norm ($\|\cdot\|_{\infty}$) and the L_2 norm, defined respectively as follows[51]:

$$L_{\infty} = \max_{0 \le n \le N} |U(y_n, t) - u(y_n, t)|,$$

$$L_2 = \sqrt{g \sum_{n=0}^{N} |U(y_n, t) - u(y_n, t)|^2},$$

Where$U(y_n, t)$ represents the exact solution and $u(y_n, t)$ denotes the numerical approximation at the spatial node y_n and time$t$.

Furthermore, the experimental order of convergence (EOC) is calculated to assess how the error decreases as the grid is refined. The EOC is determined using the following relation [52]:

$$\text{EOC} = \frac{1}{log2} log\left(\frac{L_{\infty}(2n)}{L_{\infty}(n)}\right),$$

Which provides an estimate of the convergence rate based on the computed maximum norm errors at successive mesh refinements.

Problem 1

Suppose that thenonlinear time fractional KGE[39].

$$\frac{\delta^{\beta} u}{\delta t^{\beta}} - \frac{\delta^2 u}{\delta y^2} + u^2(y,t) = f(y,t) \qquad 0 < t \le 1, 0 < y \le 1 \tag{39}$$

The piecewise defined estimated solution attained by using the projectedtechniquefor $\beta = 1.25$, over the domain $0 \le x \le 1$with $M = 100$, $n = 100$, and time step $\Delta t = 0.01$, is illustrated in Figure 2. Anevaluation between the exact and numerical solutions under the same conditions is presented in Figure 3, demonstrating close agreement between the two. The corresponding absolute error for $\beta = 1.3$, $M = 100$, and $\Delta t = 0.001$ is shown in Figure 4, highlighting the accuracy of the proposed approach. To further assess convergence properties, Table 4 presents the experimental order of convergence (EOC) where values computed along the spatial grid for $\beta = 1.5$and$\Delta t = 0.001$. The observed convergence behavior aligns well with theoretical expectations, confirming the method's validity. The absolute numerical errors at selected grid points for problem no.1 by using $\Delta t = 0.001$ and $M = 100$ are provided in Table 1. The results indicate that the proposed scheme yields significantly improved accuracy compared to the sinc-Chebyshev collocation method (Amin). In Table 2, absolute and relative errors for the proposed method are reported at $x = 0.4, 0.6, 0.8$and $t = 0.4, 0.8$, with $M = 100$, $\Delta t = 0.001$, and $\beta = 1.6$. These results further demonstrate the superior performance of the proposed approach. A detailed evaluation of the absolute errors produced by the present technique, the variational iteration method (VIM) and the (Amin) for various values of α is given in Table 3, highlighting the improved accuracy of our method across different fractional orders. The evolution of the numerical solution over time for $\beta = 1.5$, $M = 100$, and $\Delta t = 0.001$ is depicted in Figure1.

$$f(y) = \begin{cases} 0 + 1.99762y - 0.0677902y^2 + 1.27469y^3 - 6.53758y^4, & y \in [0.00,0.05) \\ -0.0342852 + y(2.69243 + y(-5.0005 + y(14.9853 - 16.8606y))), & y \in [0.05,0.10) \\ -0.574734 + y(8.18834 + y(-24.6273 + y(42.5987 - 27.6653y))), & y \in [0.10,0.15) \\ -3.23138 + y(26.1587 + y(-67.2506 + y(82.2744 - 37.7725y))), & y \in [0.15,0.20) \\ -11.2852 + y(66.8892 + y(-139.321 + y(132.042 - 46.9499y))), & y \in [0.20,0.25) \\ -29.8998 + y(141.878 + y(-244.696 + y(189.364 - 54.9713y))), & y \in [0.25,0.30) \\ -65.9286 + y(262.143 + y(-384.147 + y(251.253 - 61.6391y))), & y \in [0.30,0.35) \\ -127.266 + y(436.341 + y(-555.026 + y(314.381 - 66.7892y))), & y \in [0.35,0.40) \\ -221.743 + y(668.845 + y(-751.117 + y(375.206 - 70.2946y))), & y \in [0.40,0.45) \\ -355.606 + y(957.925 + y(-962.693 + y(430.11 - 72.0692y))), & y \in [0.45,0.50) \\ -531.676 + y(1294.19 + y(-1176.77 + y(475.538 - 72.0692y))), & y \in [0.50,0.55) \\ -747.321 + y(1659.45 + y(-1377.56 + y(508.142 - 70.2946y))), & y \in [0.55,0.60) \\ -992.425 + y(2026.07 + y(-1547.15 + y(524.916 - 66.7892y))), & y \in [0.60,0.65) \\ -1247.54 + y(2357.01 + y(-1666.26 + y(523.327 - 61.6391y))), & y \in [0.65,0.70) \\ -1482.46 + y(2606.57 + y(-1715.25 + y(501.426 - 54.9713y))), & y \in [0.70,0.75) \\ -1655.42 + y(2721.86 + y(-1675.12 + y(457.949 - 46.9499y))), & y \in [0.75,0.80) \\ -1713.15 + y(2645.08 + y(-1528.63 + y(392.389 - 37.7725y))), & y \in [0.80,0.85) \\ -1591.93 + y(2316.44 + y(-1261.41 + y(305.053 - 27.6653y))), & y \in [0.85,0.90) \\ -1218.67 + y(1676.17 + y(-862.213 + y(196.892 - 16.8606y))), & y \in [0.90,0.95) \\ -591.69 + y(771.51 + y(-375.194 + y(80.879 - 6.53758y))), & y \in [0.95,1.00] \end{cases}$$

Table 1: Exact and Approximate solution for Problem.1 $\boldsymbol{with\ \beta = 1.3\ and\ \Delta t = 0.001}$

y	Exact Solution	Approximate solution	Error
0.	1.	1.	1.01674*10^-12
0.01	0.975187	0.975191	4.04674*10^-6
0.02	0.950747	0.950755	7.7968*10^-6
0.03	0.926679	0.92669	0.0000112662
0.04	0.90298	0.902994	0.0000144701
0.05	0.879648	0.879666	0.0000174232
0.97	0.000155885	0.000162564	6.67925*10^-6
0.98	0.0000565685	0.0000621414	5.57288*10^-6
0.99	0.00001	0.0000141509	4.15091*10^-6
1.	0.	2.11758*10^-22	2.11758*10^-22

Table 2: Absolute errors for Problem 1 for$\boldsymbol{M = 100\ and\ \Delta t = 0.001}$ with different values of $\boldsymbol{\beta}$

y	Method [39]		Proposed Method	
	$\beta=1.5$	$\beta=1.9$	$\beta=1.5$	$\beta=1.9$
0.1	8.7104×10^{-4}	5.0451×10^{-4}	2.4413×10^{-6}	1.1332×10^{-6}
0.2	8.7782×10^{-4}	7.5329×10^{-5}	1.8208×10^{-8}	1.0675×10^{-6}

0.3	6.2088×10^{-4}	1.1242×10^{-4}	1.3064×10^{-6}	9.9868×10^{-6}
0.4	5.7014×10^{-4}	1.6773×10^{-4}	8.9107×10^{-7}	9.25794×10^{-6}
0.5	5.1475×10^{-4}	2.5023×10^{-4}	5.6741×10^{-6}	8.47752×10^{-6}
0.6	4.8947×10^{-4}	2.5023×10^{-4}	3.2762×10^{-6}	7.62788×10^{-6}
0.7	5.1672×10^{-4}	2.5023×10^{-4}	1.6256×10^{-6}	6.67952×10^{-6}
0.8	5.3918×10^{-4}	2.5023×10^{-4}	6.2141×10^{-6}	5.57288×10^{-6}
0.9	6.0661×10^{-4}	2.5023×10^{-5}	1.4150×10^{-6}	4.15091×10^{-6}

Table.3: Absolute and relative errors for Problem no.1 for $M = 100$ and $\Delta t = 0.001$ with value of $\beta = 1.6$

T	Y	Method [39]		Proposed method	
		L_∞	L_2	L_∞	L_2
0.4	0.4	9.3726×10^{-4}	1.3282×10^{-2}	1.0077×10^{-6}	3.1034×10^{-6}
	0.6	9.4592×10^{-4}	1.6950×10^{-2}	1.2017×10^{-6}	4.4392×10^{-6}
	0.8	6.5448×10^{-4}	1.4462×10^{-1}	2.4137×10^{-6}	6.9530×10^{-6}
0.8	0.4	1.7359×10^{-4}	8.6999×10^{-4}	5.1096×10^{-6}	1.9751×10^{-6}
	0.6	1.2080×10^{-4}	1.6683×10^{-3}	3.5474×10^{-6}	4.8718×10^{-6}
	0.8	2.4657×10^{-4}	1.9263×10^{-2}	4.1031×10^{-7}	2.5972×10^{-7}

Table 4: Comparison of absolute errors for Problem1 using $M = 100$ and $\Delta t = 0.001$with $\beta = 1.4$ or 1.6.

β	(y, t)	Method [39]	Method [29]	Proposed method
1.4	(0.10,0.10)	9.2852×10^{-3}	8.4385×10^{-4}	2.1509×10^{-7}
	(0.30,0.30)	3.5651×10^{-2}	5.3780×10^{-3}	1.0721×10^{-6} 2.1509×10^{-7}
	(0.50,0.50)	6.4449×10^{-2}	5.3227×10^{-4}	6.8403×10^{-6}
	(0.70,0.70)	9.1443×10^{-2}	1.9159×10^{-3}	2.4912×10^{-6}
	(0.90,0.90)	9.2321×10^{-4}	1.8996×10^{-3}	4.2507×10^{-7}
1.6	(0.10,0.10)	4.1518×10^{-4}	1.1685×10^{-4}	6.4258×10^{-6}
	(0.30,0.30)	1.7757×10^{-2}	2.8863×10^{-5}	5.8719×10^{-7}
	(0.50,0.50)	3.8327×10^{-2}	1.7692×10^{-5}	2.4171×10^{-6}

	(0.70,0.70)	6.1379×10^{-2}	1.4334×10^{-5}	3.3217×10^{-8}
	(0.90,0.90)	3.8618×10^{-2}	1.7449×10^{-5}	2.3148×10^{-7}

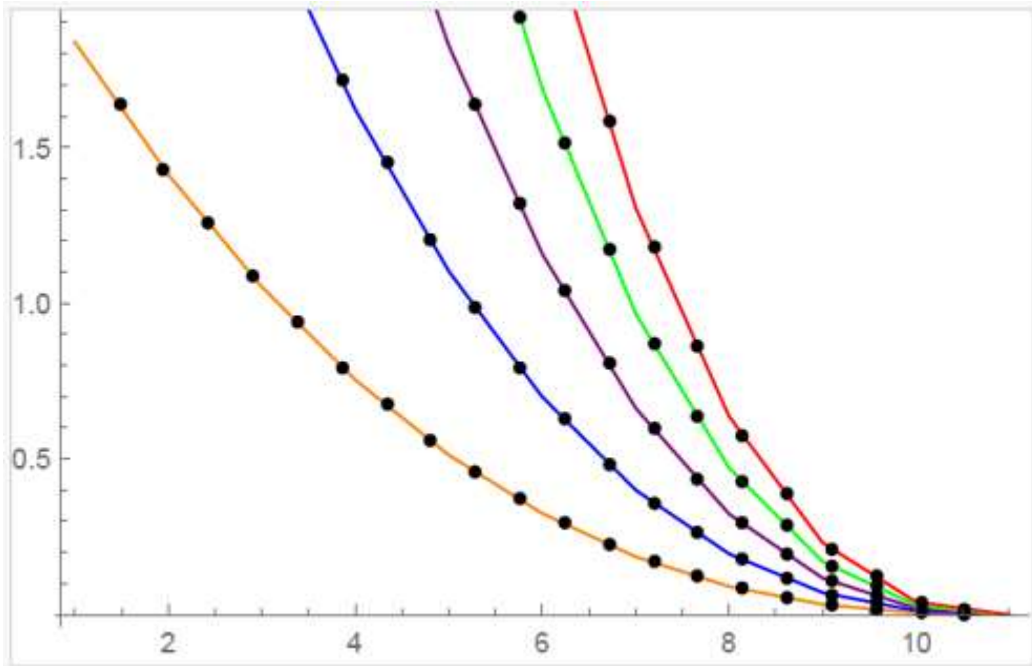

FIGURE 1: Numerical solution of Problem 1 for $\Delta t = 0.001$ and $M = 100$, with $\beta = 1.5$ at different time stages

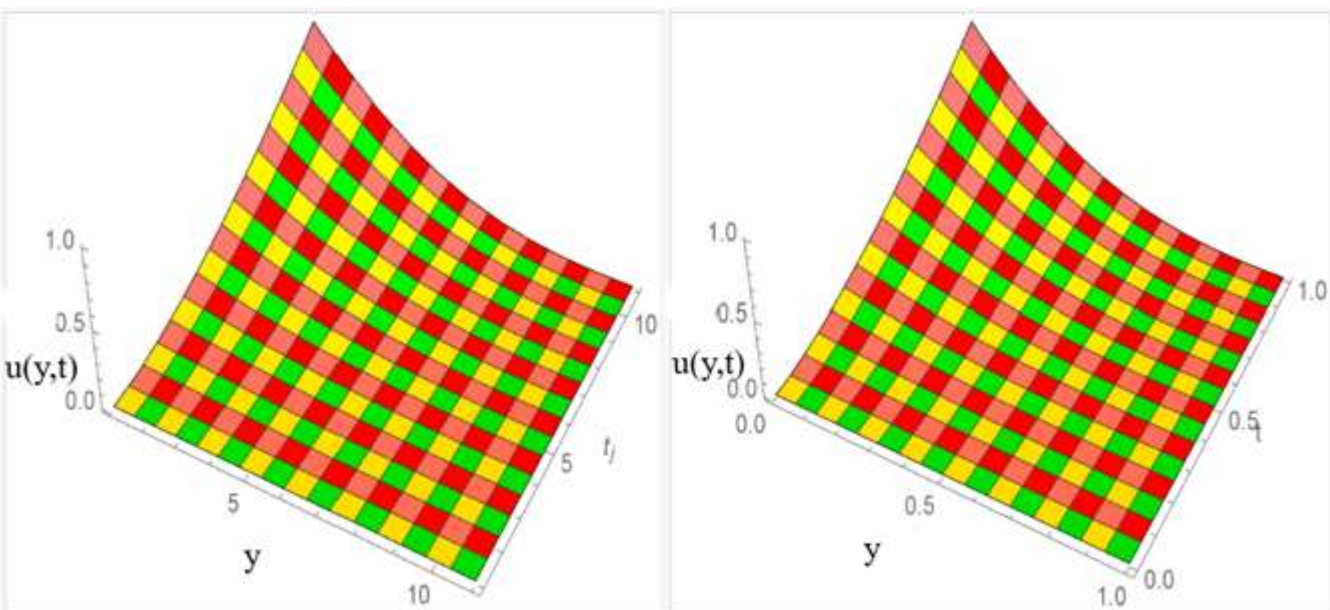

FIGURE 2: Exact and Approximate solution for Problem1 using $\Delta t = 0.001$ and $M = 100$, with $\beta = 1.5$ at different time stages.

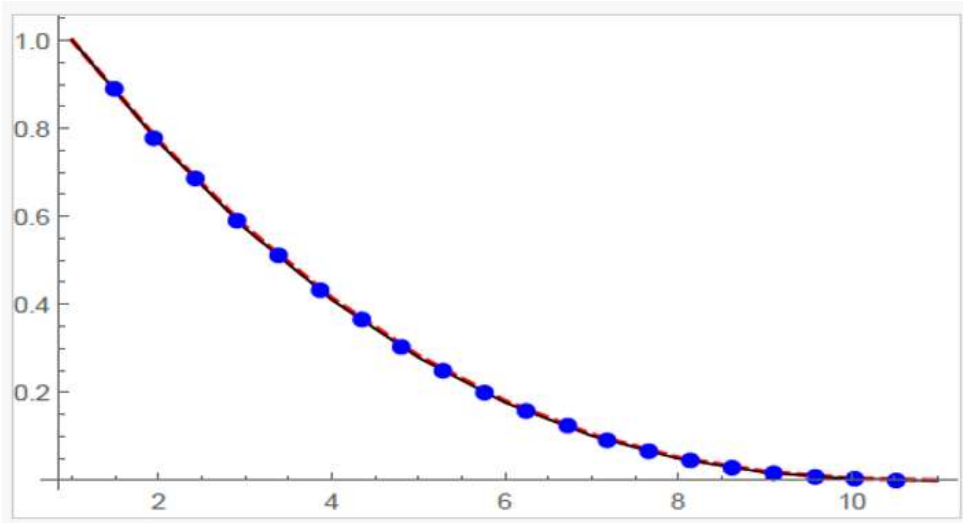

FIGURE 3: Approximate and exact solution for Problem 1 for $\Delta t = 0.001$ and $M = 100$ with $\beta = 1.5$ at different time stages.

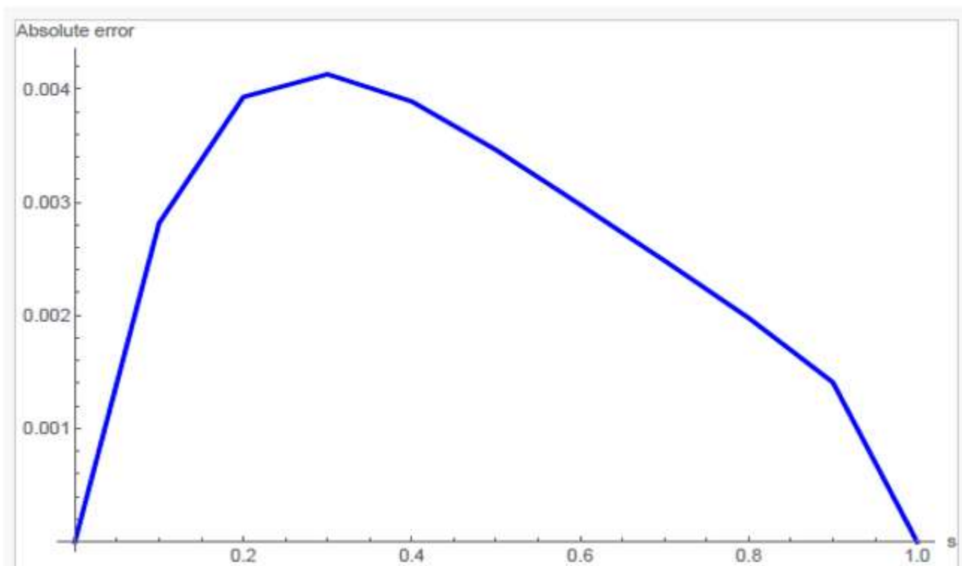

FIGURE 4: Absolute errors for Problem 1 for $\Delta t = 0.001$ and $M = 100$ with$\beta = 1.5$

Problem 2

Consider the following KGE [39]

$$\frac{\partial^\beta}{\partial t^\beta} u(y,t) - \frac{\partial^2}{\partial y^2} u(y,t) + u(y,t) + \frac{3}{2} u^3(y,t) = f(y,t), \qquad 0 < y \le 1, 0 < t \le 1 \qquad (40)$$

Where the factoring term f (y, t) on right hand side is given by

$$f(y,t) = \frac{1}{2}\eta(3+\beta)\sin(\pi y)t^2 + (1+\pi^2)t^{2+\beta}\sin(\pi y) + \frac{3}{2}[\sin(\pi y)t^{2+\beta}]^3$$

In Table 6, error norms corresponding to various values of the fractional derivative order β are reported for N = 40 and M = 1000. As observed in earlier examples, the computed error norms remain consistently low across the selected values of β, demonstrating the method's reliability. Table 7 provides a comparison of absolute errors at specific nodal points y at time $t = 1$, for fractional orders $\beta = 1.3, 1.6$, and 1.8, against the results obtained by Nagy [39]. To ensure alignment of nodal points for fair comparison, parameters were set to $N = 100$ and $M = 1000$. The results clearly indicate that the Crank–Nicolson finite difference method employed in the present work offers higher accuracy compared to the shifted Chebyshev polynomial-based method used in Nagy's study [39]. Further comparisons are provided in Tables 8 and 9, which display absolute errors for fractional orders $\beta = 1.4$ and $\beta = 1.6$, evaluated against the methods of Odibat& Momani [47], Nagy [39], Amin et al. [51], and Sahu & Jena Using a consistent setup with $N = 100$, $M = 1000$, and $t = 1$, the results demonstrate that the proposed method yields lower absolute errors at most of the selected points. Specifically, it outperforms the techniques of Odibat& Momani [47], Nagy [39], and Amin et al. [51] and exhibits improved accuracy over Sahu & Jena at the majority of the nodal points. These comparisons underline the robustness and effectiveness of the proposed numerical approach. Finally, Tables 9 and 10 present a comparative analysis of absolute error norms with those reported by Yaseen et al. and Vivas-Cortez et al. again using $N = 100$ and $M = 1000$. The results confirm that for $\beta = 1.5$, the proposed Crank–Nicolson finite difference method achieves superior accuracy compared to Yaseen et al.'s trigonometric B-spline approach. Additionally, for $\beta = 1.7$, the results obtained show close agreement with those of Vivas-Cortez et al. further validating the effectiveness of the presented method across a range of fractional orders.

Table 5: Approximate solutions for Problem 2 using different values β with $N = 40$ and $M = 1000$.

y	Exact Solution	Approximate solution	Error
0.000	0.000000	-9.22821×10^{-14}	9.22821×10^{-14}
0.025	0.0784591	0.0784485	0.0000105758
0.050	0.156434	0.156414	0.0000208249
0.075	0.233445	0.233415	0.0000304374
.625	0.923880	0.923826	0.0000536748
0.650	0.891007	0.890950	0.0000565383
0.675	0.852640	0.852581	0.0000592317
0.700	0.809017	0.808956	0.0000614815
0.975	0.0784591	0.0784485	0.0000105758
1.000	0.000000	-8.22821×10^{-17}	8.22821×10^{-17}

Table 6: The error norms for different values of fractional order β for problem no.2

T	$\beta = 1.3$		$\beta = 1.6$		$\beta = 1.8$	
	L_∞	L_2	L_∞	L_2	L_∞	L_2
0.1	6.9373×10^{-6}	4.3242×10^{-6}	7.0015×10^{-6}	5.1247×10^{-6}	8.1802×10^{-6}	7.5164×10^{-6}
0.2	4.1170×10^{-5}	2.4564×10^{-5}	5.9918×10^{-5}	2.6572×10^{-5}	6.3237×10^{-5}	3.4091×10^{-5}
0.3	3.5647×10^{-5}	1.7971×10^{-5}	4.5344×10^{-5}	2.7178×10^{-5}	5.4390×10^{-5}	3.0981×10^{-5}
0.4	8.6204×10^{-4}	6.4005×10^{-4}	9.2004×10^{-4}	7.1272×10^{-5}	9.7510×10^{-4}	7.4219×10^{-4}
0.5	2.8552×10^{-5}	1.1297×10^{-5}	3.0121×10^{-5}	2.3123×10^{-5}	4.6974×10^{-5}	3.3691×10^{-5}
0.6	5.5875×10^{-5}	3.6159×10^{-5}	5.9918×10^{-5}	4.0151×10^{-5}	6.0913×10^{-5}	5.1722×10^{-5}
0.7	4.3135×10^{-5}	2.7024×10^{-5}	5.4218×10^{-5}	3.1792×10^{-5}	5.5918×10^{-5}	4.2091×10^{-5}
0.8	6.1729×10^{-5}	3.3104×10^{-5}	7.1280×10^{-5}	3.5070×10^{-5}	8.7377×10^{-5}	4.6079×10^{-5}

0.9	9.4871×10^{-4}	5.0013×10^{-4}	9.5321×10^{-4}	4.0152×10^{-4}	9.8173×10^{-4}	5.0196×10^{-4}
1	3.9222×10^{-4}	1.0901×10^{-4}	4.8104×10^{-4}	2.1395×10^{-4}	5.8999×10^{-4}	3.3098×10^{-4}

Table 7: A comparison between absolute errors at different values of β for Problem no.2.

T	$\beta = 1.5$		$\beta = 1.7$		$\beta = 1.9$	
	SCCM [39]	**Present**	**SCCM [39]**	**Present**	**SCCM [48]**	**Present**
0.1	1.6399×10^{-3}	7.5291×10^{-4}	1.5469×10^{-3}	6.7030×10^{-4}	1.4382×10^{-3}	6.5975×10^{-4}
0.2	1.2805×10^{-3}	5.3005×10^{-4}	1.1270×10^{-3}	4.8750×10^{-4}	9.4912×10^{-4}	4.6663×10^{-4}
0.3	1.0872×10^{-3}	4.9812×10^{-4}	8.9660×10^{-4}	4.3207×10^{-4}	6.7910×10^{-4}	4.2007×10^{-4}
0.4	8.4194×10^{-4}	3.8309×10^{-5}	6.3346×10^{-4}	3.2154×10^{-5}	3.9685×10^{-4}	3.0259×10^{-5}
0.5	7.8250×10^{-4}	2.9015×10^{-5}	5.6865×10^{-4}	2.8591×10^{-5}	3.2655×10^{-4}	2.6045×10^{-5}
0.6	8.4189×10^{-4}	3.8309×10^{-5}	6.3349×10^{-4}	3.2154×10^{-5}	3.9683×10^{-4}	3.0259×10^{-5}
0.7	1.0872×10^{-3}	4.9812×10^{-4}	8.9665×10^{-4}	4.3207×10^{-4}	6.7910×10^{-4}	4.2007×10^{-4}
0.8	1.2812×10^{-3}	5.3005×10^{-4}	1.1270×10^{-3}	4.8750×10^{-4}	9.4912×10^{-4}	4.6663×10^{-4}
0.9	1.6399×10^{-3}	7.5291×10^{-4}	1.5474×10^{-3}	6.7030×10^{-4}	1.4382×10^{-3}	6.5975×10^{-4}

Table 8: A comparison between absolute errors at different points for Problem 2.

(y,t)	$\beta = 1.4$			
	(odibat&Momani2009)	**SCCM(Nagy) [39]**	**(Shau &jena)**	**Present**
(0.1,0.1)	1.0406×10^{-5}	2.3810×10^{-5}	5.84379×10^{-9}	5.3893×10^{-8}
(0.2,0.2)	1.4427×10^{-4}	5.2648×10^{-5}	2.15665×10^{-6}	4.9304×10^{-6}
(0.3,0.3)	6.7118×10^{-5}	6.0189×10^{-6}	6.47025×10^{-5}	3.2817×10^{-6}
(0.4,0.4)	3.0496×10^{-3}	6.4434×10^{-5}	6.77908×10^{-4}	4.2545×10^{-5}
(0.5,0.5)	1.6352×10^{-2}	4.0015×10^{-5}	3.89738×10^{-3}	5.0074×10^{-4}
(0.6,0.6)	4.9595×10^{-2}	1.5839×10^{-4}	1.48999×10^{-2}	4.8309×10^{-4}
(0.7,0.7)	1.0678×10^{-1}	9.1925×10^{-4}	4.10207×10^{-2}	2.0151×10^{-4}
(0.8,0.8)	1.6945×10^{-1}	2.9089×10^{-3}	8.04526×10^{-2}	7.5068×10^{-4}
(0.9,0.9)	1.7523×10^{-1}	3.8739×10^{-3}	9.42916×10^{-2}	3.2029×10^{-4}

Table 9: A comparison between absolute errors at different points Problem no.2.

Y	$\beta = 1.6$			
	(odibat&Momani2009)	**(Nagy,2017) [39]**	**(Shau&Jena,2024)**	**Present**
(0.1,0.1)	1.0405×10^{-5}	2.3806×10^{-5}	2.20656×10^{-9}	2.0473×10^{-6}
(0.2,0.2)	1.4428×10^{-4}	5.2648×10^{-5}	1.41781×10^{-7}	4.7840×10^{-6}
(0.3,0.3)	6.7118×10^{-5}	6.0189×10^{-6}	5.88596×10^{-6}	5.6735×10^{-6}
(0.4,0.4)	3.0496×10^{-3}	6.4444×10^{-5}	7.77195×10^{-5}	3.5441×10^{-5}
(0.5,0.5)	1.6355×10^{-2}	4.0015×10^{-5}	5.35288×10^{-4}	2.3879×10^{-5}
(0.6,0.6)	4.9596×10^{-2}	1.5839×10^{-4}	2.37365×10^{-3}	2.0032×10^{-5}
(0.7,0.7)	1.0678×10^{-1}	9.1925×10^{-4}	7.42469×10^{-3}	1.1057×10^{-4}
(0.8,0.8)	1.6946×10^{-1}	2.9054×10^{-3}	1.64619×10^{-2}	6.5608×10^{-4}
(0.9,0.9)	1.7525×10^{-1}	3.8736×10^{-3}	2.2292×10^{-2}	5.8774×10^{-4}

Table 10: A comparison between absolute errors at different nodal points Problem no.2.

Y	$\beta = 1.5$		
	(Yaseen. et al,2021)	**(vivas-cortezet al,2024)**	**Present**
0.1	2.2439×10^{-4}	3.7668×10^{-5}	3.5712×10^{-5}
0.2	4.4182×10^{-4}	7.0998×10^{-5}	6.1072×10^{-5}
0.3	6.3348×10^{-4}	9.6628×10^{-5}	9.5518×10^{-5}
0.4	7.6864×10^{-4}	1.1253×10^{-4}	1.0252×10^{-5}
0.5	8.1775×10^{-4}	1.1796×10^{-4}	7.0031×10^{-5}
0.6	7.6863×10^{-4}	1.1256×10^{-4}	5.7518×10^{-5}
0.7	6.3348×10^{-4}	9.6629×10^{-5}	4.6319×10^{-5}
0.8	4.4182×10^{-4}	7.0998×10^{-5}	4.3012×10^{-5}
0.9	2.2438×10^{-4}	3.7669×10^{-5}	3.0984×10^{-5}

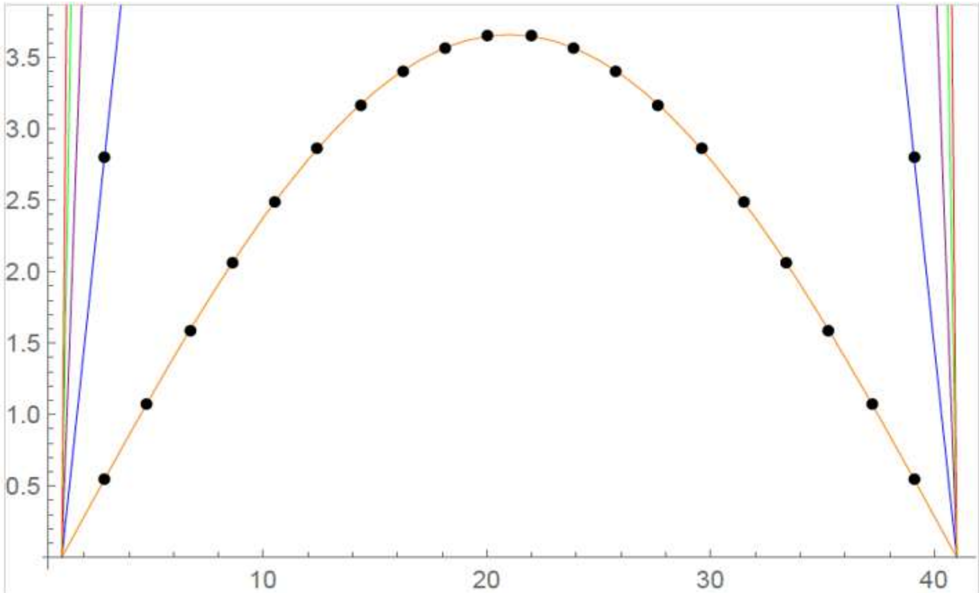

FIGURE 5: Numerical solution of Problem 2 with $M = 1000$ at different values of β.

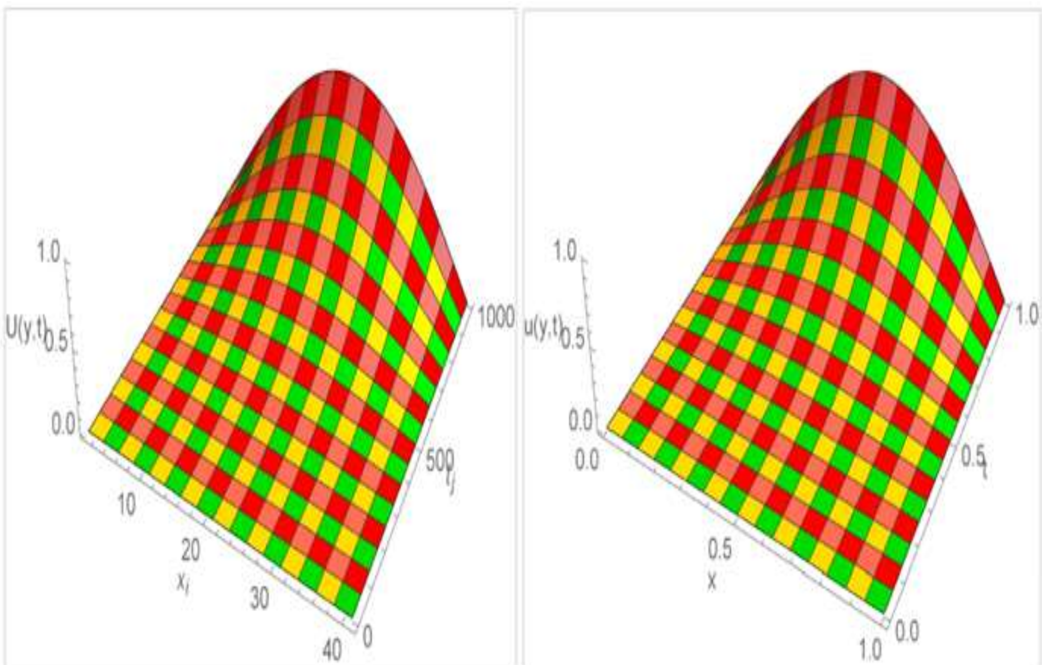

FIGURE 6: Exact and Approximate solution for Problem 2 with $M = 1000$ and different values of β.

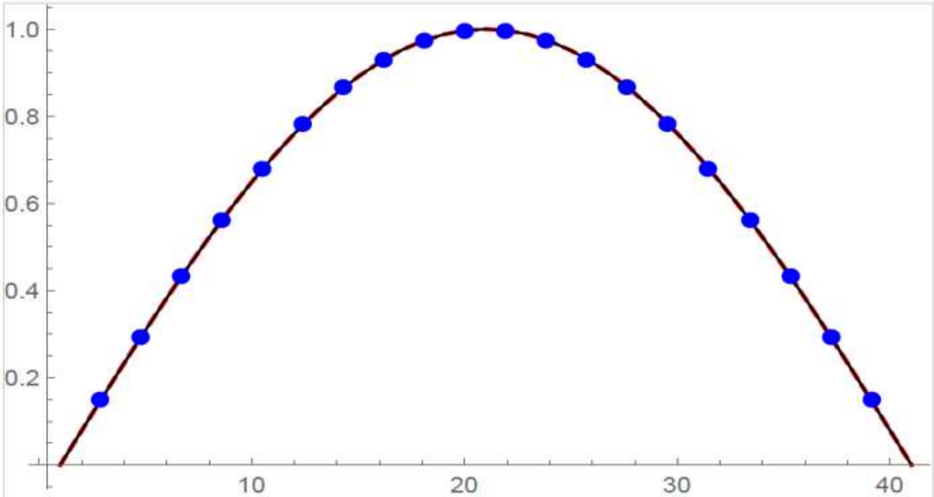

FIGURE 7: Exact and Numerical solution for Problem 2 with $M = 1000$ and different values of β.

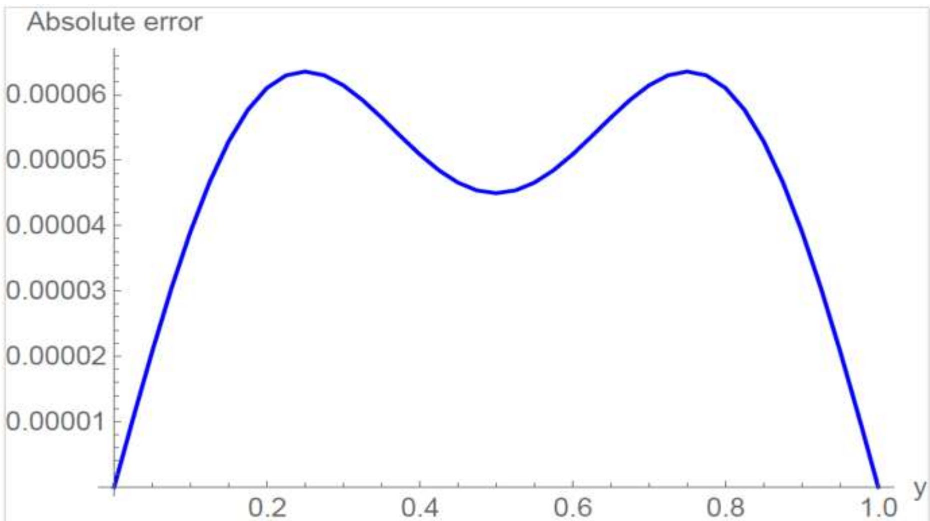

FIGURE 8: Absolute error for Problem 2 with $M = 1000$ and different values of β.

8. Conclusion

In this study, a numerical analysis of the time-fractional Klein–Gordon equation has been carried out using the redefined extended cubic B-spline (RECBS) collocation method. The temporal domain is discretized using a finite central difference approach, while the spatial domain is approximated through RECBS basis functions to build the solution curve. The proposed numerical scheme is proven to be unconditionally stable, with the spatial and temporal convergence orders theoretically established as $\mathcal{O}(h^2)$and$\mathcal{O}(\Delta t^{2-\alpha})$, respectively. The numerical results validate the expected convergence rates and confirm the method's accuracy and robustness. Furthermore, comparative assessments with existing techniques such as the Variational Iteration Method (VIM) [29] and the Sinc–Chebyshev Collocation Method (SCCM) [39] demonstrate that the proposed approach achieves superior performance in terms of accuracy and efficiency.

References

1. Babaei A, Jafari H, Liya A. Mathematical models of HIV/AIDS and drug addiction in prisons. The European Physical Journal Plus. 2020;135(5):1-12.

2. Rossikhin YA, Shitikova M. Application of fractional derivatives to the analysis of damped vibrations of viscoelastic single mass systems. Acta Mechanica. 1997;120(1):109-25.

3. Luo M-J, Milovanovic GV, Agarwal P. Some results on the extended beta and extended hypergeometric functions. Applied Mathematics and Computation. 2014;248:631-51.

4. Zhang Y. Time-fractional Klein-Gordon equation: formulation and solution using variational methods. WSEAS Trans Math. 2016;15:206-14.

5. Ruzhansky M, Cho YJ, Agarwal P, Area I. Advances in real and complex analysis with applications: Springer; 2017.

6. Owolabi KM, Hammouch Z. Mathematical modeling and analysis of two-variable system with noninteger-order derivative. Chaos: An Interdisciplinary Journal of Nonlinear Science. 2019;29(1).

7. Agarwal P, Baleanu D, Chen Y, Momani S, Machado JT, editors. Fractional calculus. ICFDA: International Workshop on Advanced Theory and Applications of Fractional Calculus Amman; 2019: Springer.

8. Yuste SB, Acedo L. An explicit finite difference method and a new von Neumann-type stability analysis for fractional diffusion equations. SIAM Journal on Numerical Analysis. 2005;42(5):1862-74.

9. Sweilam N, Nagy A. Numerical solution of fractional wave equation using Crank-Nicholson method. World Applied Sciences Journal. 2011;13:71-5.

10. Bhrawy A, Tharwat M, Yildirim A. A new formula for fractional integrals of Chebyshev polynomials: application for solving multi-term fractional differential equations. Applied Mathematical Modelling. 2013;37(6):4245-52.

11. ur Rehman M, Khan RA. The Legendre wavelet method for solving fractional differential equations. Communications in Nonlinear Science and Numerical Simulation. 2011;16(11):4163-73.

12. Bouzalmat I, editor Stochastic modeling of COVID-19 dynamics with emphasis on asymptomatic and isolated populations. International Conference on Mathematical Modeling with Modern Applications; 2024.

13. Mohebbi A, Abbaszadeh M, Dehghan M. High-order difference scheme for the solution of linear time fractional Klein–Gordon equations. Numerical Methods for Partial Differential Equations. 2014;30(4):1234-53.

14. Heydari MH, Hooshmandasl MR, Mohammadi F, Cattani C. Wavelets method for solving systems of nonlinear singular fractional Volterra integro-differential equations. Communications in Nonlinear Science and Numerical Simulation. 2014;19(1):37-48.

15. 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(2):885-902.

16. Sweilam N, Nagy A, El-Sayed AA. Second kind shifted Chebyshev polynomials for solving space fractional order diffusion equation. Chaos, Solitons & Fractals. 2015;73:141-7.

17. Ramezani M, Jafari H, Johnston S, Baleanu D. Complex B-spline collocation method for solving weakly singular Volterra integral equations of the second kind. Miskolc Mathematical Notes. 2015;16(2):1091-103.

18. Badr M, Yazdani A, Jafari H. Stability of a finite volume element method for the time-fractional advection-diffusion equation. Numerical Methods for Partial Differential Equations. 2018;34(5):1459-71.

19. Waqas M, Amin M, Mushtaq S. AN EFFICIENT HYBRID B-SPLINE APPROACH FOR NUMERICAL APPROXIMATION OF TIME-FRACTIONAL TELEGRAPH EQUATIONS. Center for Management Science Research. 2025;3(3):992-1011.

20. Atangana A, Secer A, editors. A note on fractional order derivatives and table of fractional derivatives of some special functions. Abstract and applied analysis; 2013: Wiley Online Library.

21. Srivastava H, Agarwal P. Certain fractional integral operators and the generalized incomplete hypergeometric functions. Applications and Applied Mathematics: An International Journal (AAM). 2013;8(2):1.

22. Goufo EFD. Solvability of chaotic fractional systems with 3D four-scroll attractors. Chaos, Solitons & Fractals. 2017;104:443-51.

23. Asif N, Hammouch Z, Riaz M, Bulut H. Analytical solution of a Maxwell fluid with slip effects in view of the Caputo-Fabrizio derivative⋆. The European Physical Journal Plus. 2018;133(7):272.

24. Singh J, Kumar D, Hammouch Z, Atangana A. A fractional epidemiological model for computer viruses pertaining to a new fractional derivative. Applied mathematics and computation. 2018;316:504-15.

25. Owolabi KM, Atangana A. Computational study of multi-species fractional reaction-diffusion system with ABC operator. Chaos, Solitons & Fractals. 2019;128:280-9.

26. Goufo EFD, Toudjeu IT. Analysis of recent fractional evolution equations and applications. Chaos, Solitons & Fractals. 2019;126:337-50.

27. Batiha B, Noorani M, Hashim I. Numerical solution of sine-Gordon equation by variational iteration method. Physics Letters A. 2007;370(5-6):437-40.

28. Yusufoğlu E. The variational iteration method for studying the Klein–Gordon equation. Applied Mathematics Letters. 2008;21(7):669-74.

29. Odibat Z, Momani S. The variational iteration method: an efficient scheme for handling fractional partial differential equations in fluid mechanics. Computers & Mathematics with Applications. 2009;58(11-12):2199-208.

30. Jafari H, Saeidy M, Arab FM. Solving nonlinear Klein-Gordon equation with a quadratic nonlinear term using homotopy analysis method. 2009.

31. Jafari H, Khalique CM, Ramezani M, Tajadodi H. Numerical solution of fractional differential equations by using fractional B-spline. Central European Journal of Physics. 2013;11(10):1372-6.

32. Vong S, Wang Z. A compact difference scheme for a two dimensional fractional Klein–Gordon equation with Neumann boundary conditions. Journal of Computational Physics. 2014;274:268-82.

33. Vong S, Wang Z. A high-order compact scheme for the nonlinear fractional K lein–G ordon equation. Numerical Methods for Partial Differential Equations. 2015;31(3):706-22.

34. 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(3):40.

35. 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. Computational and Applied Mathematics. 2018;37(5):6212-29.

36. Karaağaç B, Esen A, Uzunyol MH. Numerical Solutions of Time fractional Klein Gordon Equation using Crank-Nicolson Finite Difference Method. Journal of the Institute of Science and Technology. 2024;14(4):1717-30.

37. Chen H, Lü S, Chen W. A fully discrete spectral method for the nonlinear time fractional Klein-Gordon equation. Taiwanese Journal of Mathematics. 2017;21(1):231-51.

38. Lyu P, Vong S. A linearized second-order scheme for nonlinear time fractional Klein-Gordon type equations. Numerical Algorithms. 2018;78(2):485-511.

39. Nagy A. Numerical solution of time fractional nonlinear Klein–Gordon equation using Sinc–Chebyshev collocation method. Applied Mathematics and Computation. 2017;310:139-48.

40. Sahu I, Jena SR. An efficient technique for time fractional Klein-Gordon equation based on modified Laplace Adomian decomposition technique via hybridized Newton-Raphson Scheme arises in relativistic fractional quantum mechanics. Partial Differential Equations in Applied Mathematics. 2024;10:100744.

41. Yaseen M, Abbas M. An efficient cubic trigonometric B-spline collocation scheme for the time-fractional telegraph equation. Applied Mathematics-A Journal of Chinese Universities. 2020;35(3):359-78.

42. Yaseen M, Abbas M, Ahmad B. Numerical simulation of the nonlinear generalized time-fractional Klein–Gordon equation using cubic trigonometric B-spline functions. Mathematical Methods in the Applied Sciences. 2021;44(1):901-16.

43. Vivas-Cortez M, Huntul M, Khalid M, Shafiq M, Abbas M, Iqbal MK. Application of an extended cubic B-spline to find the numerical solution of the generalized nonlinear time-fractional Klein–Gordon equation in mathematical physics. Computation. 2024;12(4):80.

44. 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. Advances in Difference Equations. 2019;2019(1):514.

45. 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. Advances in Difference Equations. 2019;2019(1):378.

46. Wasim I, Abbas M, Iqbal MK. A new extended B-spline approximation technique for second order singular boundary value problems arising in physiology. J Math Comput Sci. 2019;19(4):258-67.

47. Sharifi S, Rashidinia J. Numerical solution of hyperbolic telegraph equation by cubic B-spline collocation method. Applied Mathematics and Computation. 2016;281:28-38.

48. Nauman K, Abbas M, Iqbal MK, Dumitru B. A numerical investigation of Caputo time fractional Allen–Cahn equation using redefined cubic B-spline functions. Advances in Continuous and Discrete Models. 2020;2020(1).

49. de Boor C. On the convergence of odd-degree spline interpolation. Journal of approximation theory. 1968;1(4):452-63.

50. Hall C. On error bounds for spline interpolation. Journal of approximation theory. 1968;1(2):209-18.

51. Iqbal MK, Abbas M, Zafar B. New quartic B-spline approximations for numerical solution of fourth order singular boundary value problems. Punjab University Journal of Mathematics. 2020;52(3).

52. Wasim I, Abbas M, Amin M. Hybrid B-Spline Collocation Method for Solving the Generalized Burgers-Fisher and Burgers-Huxley Equations. Mathematical Problems in Engineering. 2018;2018(1):6143934.