fractal and fractional

MDPI

Article

Comparative Numerical Study of Spline-Based Numerical Techniques for Time Fractional Cattaneo Equation in the Sense of Caputo–Fabrizio

Muhammad Yaseen [1,*], Qamar Un Nisa Arif [1], Reny George [2] and Sana Khan [1]

[1] Department of Mathematics, University of Sargodha, Sargodha 40100, Pakistan; qamararif449@gmail.com (Q.U.N.A.); sanakhanpm28@gmail.com (S.K.)
[2] Department of Mathematics, College of Science and Humanities in Al-Kharj, Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia; r.kunnelchacko@psau.edu.sa
* Correspondence: yaseen.yaqoob@uos.edu.pk

Abstract: This study focuses on numerically addressing the time fractional Cattaneo equation involving Caputo–Fabrizio derivative using spline-based numerical techniques. The splines used are the cubic B-splines, trigonometric cubic B-splines and extended cubic B-splines. The space derivative is approximated using B-splines basis functions, Caputo–Fabrizio derivative is discretized, using a finite difference approach. The techniques are also put through a stability analysis to verify that the errors do not pile up. The proposed scheme's convergence analysis is also explored. The key advantage of the schemes is that the approximation solution is produced as a smooth piecewise continuous function, allowing us to approximate a solution at any place in the domain of interest. A numerical study is performed using various splines, and the outcomes are compared to demonstrate the efficiency of the proposed schemes.

Keywords: cubic B-splines; trigonometric cubic B-splines; extended cubic B-splines; Caputo–Fabrizio derivative; Cattaneo equation

Citation: Yaseen, M.; Nisa Arif, Q.U.; George, R.; Khan, S. Comparative Numerical Study of Spline-Based Numerical Techniques for Time Fractional Cattaneo Equation in the Sense of Caputo–Fabrizio. *Fractal Fract.* **2022**, *6*, 50. https://doi.org/10.3390/fractalfract6020050

Academic Editors: Amar Debbouche and Hari Mohan Srivastava

Received: 15 December 2021
Accepted: 11 January 2022
Published: 18 January 2022

1. Introduction

The time fractional Cattaneo differential equation (TFCDE) under consideration is [1]

$$\frac{\partial v(s,t)}{\partial t} + {}_a^{CF}\mathfrak{D}_t^{\alpha} v(s,t) = \frac{\partial^2 v(s,t)}{\partial^2 s} + g(s,t), \tag{1}$$

with initial conditions

$$\begin{cases} v(s,0) = \phi(s), \\ v_t(s,0) = \psi(s), \end{cases} \qquad 0 \le s \le L, \tag{2}$$

and the boundary conditions,

$$\begin{cases} v(0,t) = f_1(t), \\ v(L,t) = f_2(t), \end{cases} \qquad t \ge 0, \tag{3}$$

where $(s,t) \in \Delta = [0,L] \times [0,T]$, $1 < \alpha < 2$, $g \in C[0,T]$, and $f_1(t), f_2(t), \phi(s), \psi(s)$ are known functions. Moreover, ${}_a^{CF}\mathfrak{D}_t^{\alpha} v(s,t)$ is the Caputo-Fabrizio derivative given by

$${}_a^{CF}\mathfrak{D}_t^{\alpha} v(s,t) = \frac{M(\alpha)}{2-\alpha} \int_a^t v''(s,x) exp[\sigma(t-x)]dx,$$

where $M(0) = M(1) = 1$ and $\sigma = \frac{1-\alpha}{2-\alpha}$.

For mathematical modeling of real-world problems, fractional differential equations are often used. Scientists in a variety of fields are pushed to improve the interpretations of their findings by utilizing the fractional order derivatives, which are particularly useful. In mathematical modeling of many scientific situations, fractional order differential equations provide more accurate information than regular differential equations. Fractional derivatives are used to describe a variety of physical phenomena [2]. This is owing to the fact that fractional operators assess both global and local properties when analyzing system evolution. In addition, integer-order calculus can sometimes contradict the experimental results; therefore, non-integer order derivatives may be preferable [3]. It is difficult to determine the solution to fractional differential equations (FDEs). As a result, a numerical method must be used to obtain the solution to these partial differential equations. To tackle these problems numerically, many approaches have been developed and extended. The existence of solution of FDEs can be seen in [4]. Diethelm et al. presented the predictor-corrector method [5] for the numerical solution of FDEs. Meerschaert and Tadjern [6] developed a finite difference method for a fractional advection–dispersion equation. The homotopy analysis method [7] for the fractional initial value problem was developed by Hashim et al. An eigenvector expansion method for motion containing fractional derivatives was presented by Suarez and Shokooh [8].

When compared to the finite difference approach, other spectral methods, such as the operational matrix method, are particularly popular since they provide good accuracy and take less time to compute. This method works well with fractional ordinary differential equations (ODEs), fractional partial differential equations (PDEs), and variable order PDEs. Jafari et al. [9] gave applications of Legendre wavelets in solving FDEs numerically. The Haar wavelet operational matrix of fractional order integration and its applications in solving fractional order differential equations can be seen in [10]. Chebyshev wavelets [11] were used by Yuanlu for solving a nonlinear fractional order differential equation. Li and Sun [12] developed a generalized block pulse operational matrix method for the solution of FDEs. Obidat [13] used Legendre polynomials to approximate the solution of nonlinear FDEs. Genocchi polynomials [14] were used by Araci to find numerical solutions of FDEs. Grbz and Sezer [15] solved a class of initial and boundary value problems arising in science and engineering using Laguerre polynomials. Caputo and Fabrizio proposed one of the most recent fractional order derivatives. For more applications of this new derivative and the related work, the reader is referred to [1,16–29].

In comparison to polynomials, the B-splines based collocation methods provide a good approximation rate, are computationally quick, numerically consistent, and have second-order continuity. To obtain numerical solutions to differential equations, multiple numerical approaches based on various forms of B-splines functions were recently utilized. Inspired by the popularity of spline approaches in finding numerical solutions of fractional partial differential equations, various splines-based numerical techniques have been developed for the numerical solution of the Cattaneo equation involving the Caputo–Fabrizio derivative. The main motivation behind this work is that to the authors' knowledge, this equation has not been solved using the B-splines basis functions. In the current work, B-splines are used to approximate the space derivative, while the Caputo–Fabrizio derivative is approximated using finite differences. Moreover, the presented schemes are tested for stability and convergence analysis.

2. Numerical Schemes

In this section, the cubic B-splines, extended cubic B-splines and the trigonometric cubic B-splines are used to develop numerical techniques for the numerical solution of time fractional Cattaneo equation (TFCE) (1).

2.1. Numerical Scheme Based on Cubic B-Splines

Let $\tau = \frac{T}{N}$ and $h = \frac{L}{M}$ be the step length in space and time direction, respectively. Set $t_m = m\tau$, $s_j = jh$, where the positive integers, N and M, are used. The knots s_j divide the

solution domain Δ equally into M equal subintervals $[s_j, s_{j+1}]$, $j = 0, 1, \ldots, M-1$, where $a = s_0 < s_1 < \cdots < s_M = b$. The approximate solution $V(s,t)$ to the exact solution $v(s,t)$ in the following form is acquired by our scheme for solving (1)

$$V(s,t) = \sum_{j=-1}^{M+1} C_j(t) B_j(s), \tag{4}$$

where $C_j(t)$ are unknowns to be found, and $B_j(s)$ [30] are cubic B-splines basis (CuBS) functions given by

$$B_j(s) = \frac{1}{6h^3} \begin{cases} (s - s_j)^3, & s \in [s_j, s_{j+1}] \\ h^3 + 3h^2(s - s_{j+1}) & \\ +3h(s - s_{j+1})^2 - 3(s - s_{j+1})^3, & s \in [s_{j+1}, s_{j+2}] \\ h^3 + 3h^2(s_{j+3} - s) & \\ +3h(s_{j+3} - s)^2 - 3(s_{j+1} - s)^3, & s \in [s_{j+2}, s_{j+3}] \\ (s_{j+4} - s)^3, & s \in [s_{j+3}, s_{j+4}] \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Here, $B_{j-1}(s), B_j(s)$ and $B_{j+1}(s)$ are survived due to the local support characteristic of the cubic B-splines so that the approximation v_j^m at the grid point (s_j, t_m) at the mth time level is given as

$$v(s_j, t^m) = v_j^m = \sum_{w=j-1}^{j+1} C_w^m(t) B_w(s). \tag{6}$$

The time-dependent unknowns $C_j^m(t)$ are found using the specified initial and boundary conditions as well as the collocation conditions on $B_j(s)$. As a result, the approximation v_j^m and its required derivatives are

$$\begin{cases} v_j^m = a_1 C_{j-1}^m + a_2 C_j^m + a_1 C_{j+1}^m, \\ (v_j^m)_s = -b_1 C_{j-1}^m + b_1 C_{j+1}^m, \\ (v_j^m)_{ss} = c_1 C_{j-1}^m + c_2 C_j^m + c_1 C_{j+1}^m, \end{cases} \tag{7}$$

where $a_1 = \frac{1}{6}$, $a_2 = \frac{4}{6}$, $b_1 = \frac{1}{2h}$, $c_1 = \frac{1}{h^2}$, and $c_2 = -\frac{2}{h^2}$. Let $g = \{g^m : 0 \le m \le N\}$ be the collection of grid functions on a uniform mesh of the interval $[0, T]$ such that $\delta_t g^m = \frac{g^m - g^{m-1}}{\tau}$. A discrete approximation to ${}_0^{CF}\mathfrak{D}_t^\alpha v(s,t)$ at $(s_j, t_{m+\frac{1}{2}})$ can be obtained as [1]

$${}_0^{CF}\mathfrak{D}_t^\alpha v(s_j, t_{m+\frac{1}{2}}) = \frac{1}{(1-\alpha)\tau}(M_0 \delta_t v_j^{m+1} - \sum_{l=1}^{m}(M_{m-l} - M_{m-l+1})\delta_t v_j^l - M_m \psi_j) + R_j^{m+\frac{1}{2}}, \tag{8}$$

where,

$$M_j = \exp(\frac{1-\alpha}{2-\alpha}\tau j) - \exp(\frac{1-\alpha}{2-\alpha}\tau(j+1)), \tag{9}$$

and

$$|R_i^{m+\frac{1}{2}}| = O(\tau^2).$$

Lemma 1 ([1]). *From the definition of M_j in (9), we have $M_j > 0$ and $M_{j+1} < M_j$, $\forall j \le m$.*

Lemma 2 ([1]). *Suppose that $v(t) \in C_{s,t}^{4,4}([0, L] \times [0, T])$, then*

$$0 \le M_j \le C\tau$$

and

$$0 \leq M_j - M_{j+1} \leq C\tau M_j.$$

Now, we employ the Caputo–Fabrizio fractional derivative and CuBS to establish the numerical scheme for solving (1). Using CuBS and the approximation given in (8), we obtain

$$(v_j^m)_t + \frac{1}{(\alpha-1)\tau}(M_0\delta_t v_j^{m+1} - \sum_{l=1}^{m}(M_{m-l} - M_{m-l+1})\delta_t v_j^l - M_m\psi_j) = (v_j^{m+1})_{ss} + g_j^{m+1} + R^{m+1}, \tag{10}$$

where $R^{m+1} = O(\tau^2 + h^2)$. Thus, by ignoring R^{m+1} and using the discretization $(v_j^m)_t = \frac{v_j^{m+1} - v_j^m}{\tau}$, we have

$$\begin{aligned}(\alpha-1)(v_j^{m+1} - v_j^m) + \frac{M_0}{\tau}(v_j^{m+1} - v_j^m) - \frac{1}{\tau}\sum_{l=1}^{m}(M_{m-l} - M_{m-l+1})(v_j^l - v_j^{l-1})\\ - M_m\psi_j) = (\alpha-1)(v_j^{m+1})_{ss} + (\alpha-1)\tau g_j^{m+1}.\end{aligned}$$

Rearranging the above equation, we obtain

$$\sigma v_j^{m+1} - \mu(v_j^{m+1})_{ss} = \sigma v_j^m + \frac{1}{\tau}\sum_{l=1}^{m}(M_{m-l} - M_{m-l+1})(v_j^l - v_j^{l-1}) + M_m\psi_j + \mu g_j^{m+1}, \tag{11}$$

where $\sigma = (\alpha - 1 + \frac{M_0}{\tau})$ and $\mu = (\alpha-1)\tau$. Using the CuBS approximation (7) in (11), we obtain

$$\begin{aligned}\eta_1 C_{j-1}^{m+1} + \eta_2 C_j^{m+1} + \eta_1 C_{j+1}^{m+1} &= \eta_3 C_{j-1}^m + \eta_4 C_j^m + \eta_3 C_{j+1}^m\\ &+ \frac{1}{\tau}\sum_{l=1}^{m}(M_{m-l} - M_{m-l+1})[(a_1 C_{j-1}^l + a_2 C_j^l + a_1 C_{j+1}^l)\\ &- (a_1 C_{j-1}^{l-1} + a_2 C_j^{l-1} + a_1 C_{j+1}^{l-1})] + M_m\psi_j + \mu g_j^{m+1},\end{aligned} \tag{12}$$

where $\eta_1 = \sigma a_1 - \mu c_1$, $\eta_2 = \sigma a_2 - \mu c_2$, $\eta_3 = \sigma a_1$,and $\eta_4 = \sigma a_2$. In matrix notation, the above equation is expressed as

$$A_1 C^{m+1} = A_2 C^m + B_1(\frac{1}{\tau}\sum_{l=1}^{m}(M_{m-l} - M_{m-l+1})(C^l - C^{l-1})) + M_m\Psi_+\mu G,$$

where the matrices A_1, A_2, B_1, Ψ and G are

$$A_1 = \begin{bmatrix}\eta_1 & \eta_2 & \eta_1 & 0 & \dots & 0\\ 0 & \eta_1 & \eta_2 & \eta_1 & \ddots & \vdots\\ \vdots & \ddots & \ddots & \ddots & \ddots & 0\\ 0 & \dots & \eta_1 & \eta_2 & \eta_1 & 0\\ 0 & \dots & 0 & \eta_1 & \eta_2 & \eta_1\end{bmatrix},$$

$$A_2 = \begin{bmatrix}\eta_3 & \eta_4 & \eta_3 & 0 & \dots & 0\\ 0 & \eta_3 & \eta_4 & \eta_3 & \ddots & \vdots\\ \vdots & \ddots & \ddots & \ddots & \ddots & 0\\ 0 & \dots & \eta_3 & \eta_4 & \eta_3 & 0\\ 0 & \dots & 0 & \eta_3 & \eta_4 & \eta_3\end{bmatrix},$$

$$B_1 = \begin{bmatrix} a_1 & a_2 & a_1 & 0 & \dots & 0 \\ 0 & a_1 & a_2 & a_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & a_1 & a_2 & a_1 & 0 \\ 0 & \dots & 0 & a_1 & a_2 & a_1 \end{bmatrix},$$

$$\Psi = \begin{bmatrix} \psi_0^{m+1}, & \psi_1^{m+1} & ,\dots, & \psi_M^{m+1} \end{bmatrix}^T,$$

and

$$G = \begin{bmatrix} g_0^{m+1}, & g_1^{m+1} & ,\dots, & g_M^{m+1} \end{bmatrix}^T.$$

The above system gives $(M+1)$ equations in $(M+3)$ unknowns. For a unique solution, two additional linear equations are necessary. For this purpose, the boundary conditions are utilized as

$$\begin{cases} a_1 C_{-1}^{m+1} + a_2 C_0^{m+1} + a_1 C_1^{m+1} = f_1(t_{m+1}), \\ a_1 C_{M-1}^{m+1} + a_2 C_M^{m+1} + a_1 C_{M+1}^{m+1} = f_2(t_{m+1}). \end{cases} \tag{13}$$

By combining Equations (12) and (13), we have $(M+3) \times (M+3)$, a system of linear equations which can be solved uniquely.

2.2. Initial State

First of all, it is essential to find the initial vector $C^0 = \begin{bmatrix} C_{-1}^0, & C_0^0, & \dots & , C_M^0, & C_{M+1}^0 \end{bmatrix}^T$ to initiate the iteration procedure. This vector is obtained from initial conditions as

$$\begin{cases} v_0' & = \phi'(s_0), \\ v_j^0 & = \phi(s_j), \quad j = 0, 1, 2, 3, \dots, M, \\ v_M' & = \phi'(s_M). \end{cases}$$

Thus, $(M+3) \times (M+3)$ a system of linear equations results, and this system can be written in matrix notation as

$$A_3 C^0 = B_2,$$

where the matrices A_3, C^0 and B_2 are

$$A_3 = \begin{bmatrix} -b_1 & 0 & b_1 & 0 & \dots & 0 \\ a_1 & a_2 & a_1 & 0 & \dots & 0 \\ 0 & a_1 & a_2 & a_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & a_1 & a_2 & a_1 \\ 0 & \dots & 0 & -b_1 & 0 & b_1 \end{bmatrix},$$

$$C^0 = \begin{bmatrix} C_{-1}^0, & C_0^0 & ,\dots, & C_M^0, & C_{M+1}^0 \end{bmatrix}^T,$$

and

$$B_2 = \begin{bmatrix} \phi'(s_0), & \phi(s_0) & ,\dots, & \phi(s_M), & \phi'(s_M) \end{bmatrix}^T.$$

2.3. Numerical Scheme Based on Extended Cubic B-Splines

A cubic B-spline of degree four with a free parameter η is called an extended cubic B-spline. This kind of cubic B-spline was introduced by Han and Liu in 2003. We follow the same notations for the time and space discretizations that we used before. The extended cubic B-spline (ECuBS) basis functions, $B_j^4(s,\eta)$ are given by

$$B_j^4(s,\eta) = \frac{1}{24h^4}\begin{cases} 4h(1-\eta)(s-s_j)^3 + 3\eta(s-s_j)^4, & s \in [s_j, s_{j+1}] \\ (4-\eta)h^4 + 12h^3(s-s_{j+1}) + 6h^2(2+\eta)(s-s_{j+1})^2 & \\ -12h(s-s_{j+1})^3 - 3\eta(s-s_{j+1})^4, & s \in [s_{j+1}, s_{j+2}] \\ (4-\eta)h^4 + 12h^3(s_{j+3}-s) + 6h^2(2+\eta)(s_{j+3}-s)^2 & \\ -12h(s_{j+3}-s)^3 - 3\eta(s_{j+1}-s)^4, & s \in [s_{j+2}, s_{j+3}] \\ 4h(1-\eta)(s_{j+4}-s)^3 + 3\eta(s_{j+4}-s)^4, & s \in [s_{j+3}, s_{j+4}] \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

where $\eta \in [-8, 1]$. Here, $B_{j-1}^4(s), B_j^4(s)$ and $B_{j+1}^4(s)$ are survived due to local support characteristic of the cubic B-splines so that the approximation v_j^m at the grid point (s_j, t_m) at mth time level is given as

$$v(s_j, t^m) = v_j^m = \sum_{w=j-1}^{j+1} C_w^m(t) B_w^4(s,\eta). \tag{15}$$

The time-dependent unknowns $C_j^m(t)$ are found using the specified initial and boundary conditions as well as the collocation conditions on $B_j(s)$. As a result, the approximation v_j^m and its required derivatives are

$$\begin{cases} v_j^m = \omega_1 C_{j-1}^m + \omega_2 C_j^m + \omega_1 C_{j+1}^m, \\ (v_j^m)_s = -\omega_3 C_{j-1}^m + \omega_4 C_j^m + \omega_3 C_{j+1}^m, \\ (v_j^m)_{ss} = \omega_5 C_{j-1}^m + \omega_6 C_j^m + \omega_5 C_{j+1}^m, \end{cases} \tag{16}$$

where $\omega_1 = \frac{4-\eta}{24}$, $\omega_2 = \frac{8+\eta}{12}$, $\omega_3 = \frac{1}{2h}$, $\omega_4 = 0$, $\omega_5 = \frac{2+\eta}{2h^2}$ and $\omega_6 = -\frac{2+\eta}{2h^2}$. By following the same procedure as was done for cubic B-splines and using the ECuBS approximation given in (16), we obtain the following approximation to the solution of (1)

$$\begin{aligned} \eta_5 C_{j-1}^{m+1} + \eta_6 C_j^{m+1} + \eta_5 C_{j+1}^{m+1} &= \eta_7 C_{j-1}^m + \eta_8 C_j^m + \eta_7 C_{j+1}^m \\ &+ \frac{1}{\tau}\sum_{l=1}^{m}(M_{m-l} - M_{m-l+1})[(\omega_1 C_{j-1}^l + \omega_2 C_j^l + \omega_1 C_{j+1}^l) \\ &- (\omega_1 C_{j-1}^{l-1} + \omega_2 C_j^{l-1} + \omega_1 C_{j+1}^{l-1})] + M_m \psi_j + \mu g_j^{m+1}, \end{aligned} \tag{17}$$

where, $\eta_5 = \sigma\omega_1 - \mu\omega_5$, $\eta_6 = \sigma\omega_2 - \mu\omega_5$, $\eta_7 = \sigma\omega_1$ and $\eta_8 = \sigma\omega_2$. In matrix notation, the above Equation (17) is expressed as

$$A_4 C^{m+1} = A_5 C^m + B_3(\frac{1}{\tau}\sum_{l=1}^{m}(M_{m-l} - M_{m-l+1})(C^l - C^{l-1})) + M_m \Psi_+ \mu G,$$

where the matrices A_4, A_5 and B_3 are

$$A_4 = \begin{bmatrix} \eta_5 & \eta_6 & \eta_5 & 0 & \dots & 0 \\ 0 & \eta_5 & \eta_6 & \eta_5 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \eta_5 & \eta_6 & \eta_5 & 0 \\ 0 & \dots & 0 & \eta_5 & \eta_6 & \eta_5 \end{bmatrix},$$

$$A_5 = \begin{bmatrix} \eta_7 & \eta_8 & \eta_7 & 0 & \dots & 0 \\ 0 & \eta_7 & \eta_8 & \eta_7 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \eta_7 & \eta_8 & \eta_7 & 0 \\ 0 & \dots & 0 & \eta_7 & \eta_8 & \eta_7 \end{bmatrix},$$

and

$$B_3 = \begin{bmatrix} \omega_1 & \omega_2 & \omega_1 & 0 & \dots & 0 \\ 0 & \omega_1 & \omega_2 & \omega_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \omega_1 & \omega_2 & \omega_1 & 0 \\ 0 & \dots & 0 & \omega_1 & \omega_2 & \omega_1 \end{bmatrix}.$$

The above system gives $(M+1)$ equations in $(M+3)$ unknowns. For a unique solution, two additional linear equations are necessary. From the boundary conditions, we obtain the required equations as follows

$$\begin{cases} \omega_1 C_{-1}^{m+1} + \omega_2 C_0^{m+1} + \omega_1 C_1^{m+1} = f_1(t_{m+1}), \\ \omega_1 C_{M-1}^{m+1} + \omega_2 C_M^{m+1} + \omega_1 C_{M+1}^{m+1} = f_2(t_{m+1}). \end{cases} \tag{18}$$

By combining Equations (17) and (18), we have $(M+3) \times (M+3)$, a system of linear equations, which can be solved uniquely.

2.4. Numerical Scheme Based on Trigonometric Cubic B-Splines

We follow the same notations for the time and space discretizations used before. The trigonometric cubic B-spline (TCuBS) basis functions are given by [31]

$$TB_j^4(s) = \frac{1}{p} \begin{cases} l^3(s_j), & s \in [s_j, s_{j+1}) \\ l(s_j)(l(s_j)m(s_{j+2}) + m(s_{j+3})l(s_{j+1})) + m(s_{j+4})l^2(s_{j+1}), & s \in [s_{j+1}, s_{j+2}) \\ m(s_{j+4})(l(s_{j+1})m(s_{j+3}) + m(s_{j+4})l(s_{j+2})) + l(s_j)m^2(s_{j+3}), & s \in [s_{j+2}, s_{j+3}) \\ m^3(s_{j+4}), & s \in [s_{j+3}, s_{j+4}), \end{cases} \tag{19}$$

where $l(s_j) = \sin(\frac{s-s_j}{2}), m(s_j) = \sin(\frac{s_j-s}{2})$ and $p = \sin(\frac{h}{2})\sin(h)\sin(\frac{3h}{2})$.

Here, $TB_{j-1}^4(s), TB_j^4(s)$ and $TB_{j+1}^4(s)$ are survived due to the local support characteristic of the trigonometric cubic B-splines so that the approximation v_j^m at the grid point (s_j, t_m) at mth time level is given as

$$v(s_j, t^m) = v_j^m = \sum_{w=j-1}^{j+1} C_w^m(t) TB_w^4(s). \tag{20}$$

The time-dependent unknowns $C_j^m(t)$ are found using the specified initial and boundary conditions as well as the collocation conditions on $B_j(s)$. As a result, the approximation v_j^m and its required derivatives are

$$\begin{cases} v_j^m = \zeta_1 C_{j-1}^m + \zeta_2 C_j^m + \zeta_1 C_{j+1}^m, \\ (v_j^m)_s = -\zeta_3 C_{j-1}^m + \zeta_4 C_{j+1}^m + \zeta_3 C_{j+1}^m, \\ (v_j^m)_{ss} = \zeta_5 C_{j-1}^m + \zeta_6 C_j^m + \zeta_5 C_{j+1}^m, \end{cases} \tag{21}$$

where

$$\begin{cases} \zeta_1 = \csc(h)\csc(\frac{3h}{2})\sin^2(\frac{h}{2}), \\ \zeta_2 = \frac{2}{1+2\cos(h)}, \\ \zeta_3 = \frac{3}{4}\csc(\frac{3h}{2}), \\ \zeta_4 = 0, \\ \zeta_5 = \frac{3+9\cos(h)}{4\cos(\frac{h}{2})} - 4\cos(\frac{5h}{2}), \\ \zeta_6 = -\frac{3\cot^2(\frac{h}{2})}{2+4\cos(h)}. \end{cases}$$

By following the same procedure as was done for cubic B-splines and using the approximation (8) in (1), we obtain

$$\sigma v_j^{m+1} - \mu (v_j^{m+1})_{ss} = \sigma v_j^m + \frac{1}{\tau}\sum_{l=1}^{m}(M_{m-l} - M_{m-l+1})(v_j^l - v_j^{l-1}) + M_m \psi_j + \mu g_j^{m+1}, \tag{22}$$

where $\sigma = (\alpha - 1 + \frac{M_0}{\tau})$ and $\mu = (\alpha - 1)\tau$. Using the CuTBS approximation given in (21), we obtain the following approximation to the solution of (1)

$$\begin{aligned} \eta_9 C_{j-1}^{m+1} + \eta_{10} C_j^{m+1} + \eta_9 C_{j+1}^{m+1} &= \eta_{11} C_{j-1}^m + \eta_{12} C_j^m + \eta_{11} C_{j+1}^m \\ &+ \frac{1}{\tau}\sum_{l=1}^{m}(M_{m-l} - M_{m-l+1})[(\zeta_1 C_{j-1}^l + \zeta_2 C_j^l + \zeta_1 C_{j+1}^l) \\ &- (\zeta_1 C_{j-1}^{l-1} + \zeta_2 C_j^{l-1} + \zeta_1 C_{j+1}^{l-1})] + M_m \psi_j + \mu g_j^{m+1}, \end{aligned} \tag{23}$$

where $\eta_9 = \sigma\zeta_1 - \mu\zeta_5$, $\eta_{10} = \sigma\zeta_2 - \mu\zeta_6$, $\eta_{11} = \sigma\zeta_1$, and $\eta_{12} = \sigma\zeta_2$. In matrix notation, (23) is expressed as

$$A_7 C^{m+1} = A_8 C^m + B_5(\frac{1}{\tau}\sum_{l=1}^{m}(M_{m-l} - M_{m-l+1})(C^l - C^{l-1})) + M_m \Psi_+ \mu G,$$

where the matrices A_7, A_8 and B_5 are

$$A_7 = \begin{bmatrix} \eta_9 & \eta_{10} & \eta_9 & 0 & \dots & 0 \\ 0 & \eta_9 & \eta_{10} & \eta_9 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \eta_9 & \eta_{10} & \eta_9 & 0 \\ 0 & \dots & 0 & \eta_9 & \eta_{10} & \eta_9 \end{bmatrix},$$

$$A_8 = \begin{bmatrix} \eta_{11} & \eta_{12} & \eta_{11} & 0 & \dots & 0 \\ 0 & \eta_{11} & \eta_{12} & \eta_{11} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \eta_{11} & \eta_{12} & \eta_{11} & 0 \\ 0 & \dots & 0 & \eta_{11} & \eta_{12} & \eta_{11} \end{bmatrix},$$

and

$$B_5 = \begin{bmatrix} \zeta_1 & \zeta_2 & \zeta_1 & 0 & \dots & 0 \\ 0 & \zeta_1 & \zeta_2 & \zeta_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \zeta_1 & \zeta_2 & \zeta_1 & 0 \\ 0 & \dots & 0 & \zeta_1 & \zeta_2 & \zeta_1 \end{bmatrix}.$$

The above system gives $(M+1)$ equations in $(M+3)$ unknowns. For a unique solution, two additional linear equations are necessary. From the boundary conditions, we obtained these equations as follows

$$\begin{cases} \zeta_1 C_{-1}^{m+1} + \zeta_2 C_0^{m+1} + \zeta_1 C_1^{m+1} = f_1(t_{m+1}), \\ \zeta_1 C_{M-1}^{m+1} + \zeta_2 C_M^{m+1} + \zeta_1 C_{M+1}^{m+1} = f_2(t_{m+1}). \end{cases} \tag{24}$$

By combining Equations (23) and (24), we have $(M+3) \times (M+3)$, a system of linear equations, which can be solved uniquely.

2.5. Stability Analysis

This section deals with stability analysis of the scheme based on cubic B-splines. The stability analysis of the schemes based on extended and cubic trigonometric B-splines can be carried out by a similar argument. We use the Fourier method to study the stability analysis of the scheme. Let $\tilde{V}^0$ be the perturbation vector of initial values V^0 and $\tilde{V}^m$, $1 \le m \le N-1$ be the approximate solution of the scheme (12). The error vector δ^m is defined as

$$\delta^m = V^m - \tilde{V}^m, \quad 0 \le m \le N-1, \tag{25}$$

where,

$$V^m = \left[V_1^m, \quad V_2^m, \quad \dots, \quad V_{M-1}^m\right]^T,$$
$$\tilde{V}^m = \left[\tilde{V}_1^m, \quad \tilde{V}_2^m, \quad \dots, \quad \tilde{V}_{M-1}^m\right]^T,$$

and

$$\delta_j^m = V_j^m - \tilde{V}_j^m = \left[\delta_1^m, \quad \delta_2^m, \quad \dots, \quad \delta_{M-1}^m\right]^T.$$

Define the grid functions as follows:

$$\delta^m(s) = \begin{cases} \delta_j^m, \ s_j - \dfrac{h}{2} < s < s_j + \dfrac{h}{2}, \\ 0, \ 0 \le s \le \dfrac{h}{2} \text{ or } L - \dfrac{h}{2} < s < L. \end{cases}$$

We can expand $\delta^m(s)$ into Fourier series as

$$\delta^m(s) = \sum_{l=-\infty}^{\infty} d_m(l) \exp\left(\frac{I2\pi ls}{L}\right),$$

where,

$$d_m(l) = \frac{1}{L} \int_0^L \delta^m(s) \exp\left(\frac{-I2\pi ls}{L}\right) ds.$$

Denoting

$$\|\delta^m\|_2 = (\int_0^L \|\delta^m(s)\|^2 ds)^{\frac{1}{2}},$$

and using the Parseval's equality,

$$\int_0^L \|\delta^m(s)\|^2 ds = \sum_{l=-\infty}^{\infty} \|d_m(l)\|^2,$$

we obtain

$$\|\delta^m(s)\|^2 = \sum_{l=-\infty}^{\infty} \|d_m(l)\|^2.$$

We can expand δ_j^m into Fourier series, and because the difference equations are linear, we can analyze the behavior of total error by tracking the behavior of an arbitrary nth component. Based on the above analysis, we can suppose that the solution of (11) has the following form

$$\delta_j^m = d_m \exp(I\sigma_s jh), \tag{26}$$

where $\sigma_s = \frac{2\pi l}{L}$, $I = \sqrt{-1}$. Substituting the above expression into (11), we obtain

$$\sigma(\delta^{m+1}) - \mu(\delta_{ss}^{m+1}) = \sum_{l=1}^{m}(M_{m-l} - M_{m-l+1})\delta_t(\delta^l) + \sigma(\delta^m). \tag{27}$$

Using the CuBS approximation given in (7) and Equation (26) in the above equation, we obtain

$$\begin{aligned}
&d_{m+1}(\sigma(a_1\exp(-I\sigma_s h) + a_2 + a_1\exp(I\sigma_s h) - \mu(c_1\exp(-I\sigma_s h) + c_2 + c_1\exp(I\sigma_s h)))\\
&= \sum_{l=1}^{m}(M_{m-l} - M_{m-l+1})\delta_t d_l(a_1\exp(-I\sigma_s h) + a_2 + a_1\exp(I\sigma_s h)) + \sigma d_m(a_1\exp(-I\sigma_s h))\\
&+ a_2 + a_1\exp(I\sigma_s h)),\\
&\Rightarrow\ d_{m+1}(\sigma(a_2 + a_1(2\cos(\sigma_s h)) - \mu(c_2 + c_1(2\cos(\sigma_s h)))\\
&= \sum_{l=1}^{m}(M_{m-l} - M_{m-l+1})\delta_t d_l(a_2 + a_1(2\cos(\sigma_s h)) + \sigma d_m(a_2 + a_1(2\cos(\sigma_s h)),
\end{aligned}$$

which, on further simplification, reduces to

$$d_{m+1} = \frac{1}{\sigma - \mu r}\sum_{l=1}^{m}(M_{m-l} - M_{m-l+1})(\frac{d_l - d_{l-1}}{\tau}) + \frac{\sigma}{\sigma - \mu r}d_m,\ 1 \le m \le M-1, \tag{28}$$

where, $r = (\frac{c_2 + 2c_1\cos(\sigma_s h)}{a_2 + 2a_1\cos(\sigma_s h)})$.

Definition 1 ([32,33]). *A scheme is called stable if there exists a positive number C, independent of j and m such that*

$$||V^n - \tilde{V}^n|| \le C||V^0 - \tilde{V}^0||,$$

where V^n and $\tilde{V}^n$ are the exact solutions of the difference scheme and its perturbed equation, respectively.

Theorem 1. *Suppose that d_m, $(1 \le m \le N-1)$ are defined by (28), then for $\alpha \in (1,2)$, we have*

$$|d_m| \le (1 + 2C\tau)^m |d_0|, \quad m = 1, 2, \ldots, M-1.$$

Proof. We use the mathematical induction for proof. For $m = 1$, we have from (28),

$$|d_1| = |\frac{\sigma}{\sigma - \mu r}||d_0| \leq (1 + 2C\tau)|d_0|.$$

Now, suppose that

$$|d_m| \leq (1 + 2C\tau)^m |d_0|, \quad m = 1, 2, \ldots, M-2.$$

Then, by using Lemmas 1 and 2, we obtain

$$\begin{aligned} |d_{m+1}| &\leq \frac{C\tau}{\sigma - \mu r} \sum_{l=1}^{m} |(d_l - d_{l-1})| + \frac{\sigma}{\sigma - \mu r}|d_m| \\ &= \frac{C\tau}{\sigma - \mu r}|(d_m - d_0)| + \frac{\sigma}{\sigma - \mu r}|d_m| \\ &\leq \frac{2C'\tau + \sigma}{\sigma - \mu r}(1 + 2C\tau)^m |d_0| \leq (1 + 2C\tau)^{m+1}|d_0|. \end{aligned}$$

This completes the proof. □

Theorem 2. *The scheme (12) is unconditionally stable for* $\alpha \in (1, 2)$.

Proof. By using Theorem 1, Parseval's equality and $m\tau \leq T$, we obtain

$$\begin{aligned} \|V^m - \tilde{V}^m\|_{l_2}^2 &= \sum_{-\infty}^{\infty} \|d_m(l)\|^2 \\ &\leq (1 + 2C\tau)^{2m} \sum_{l=-\infty}^{\infty} \|d_0(l)\|^2 \\ &= (1 + 2C\tau)^{2m} \|\delta^0(l)\|_{l_2}^2 \\ &\leq \exp(4C\tau m)\|V^0 - \tilde{V}^0\|_{l_2}^2. \end{aligned}$$

so that

$$\|V^m - \tilde{V}^m\|_{l_2} \leq \exp(2\sqrt{C\tau})\|V^0 - \tilde{V}^0\|_{l_2}.$$

which means that the scheme is unconditionally stable. □

3. Convergence Analysis

The convergence of the scheme based on cubic B-splines is presented in this section. The convergence analysis of the extended and cubic trigonometric B-splines based numerical scheme follows accordingly. Let $e_j^m = v_j^m - V_j^m$, $1 \leq j \leq M-1$, $1 \leq m \leq N-1$ and

$$\begin{aligned} e^m &= (e_1^m, e_2^m, \ldots, e_{M-1}^m), \\ \mathbf{R}^m &= (R_1^m, R_2^m, \ldots, R_{M-1}^m), \quad 0 \leq m \leq N-1. \end{aligned}$$

From Equation (11) and $R_j^{m+1} = O(\tau^2 + h^2)$ and noting that $e_j^0 = 0$, we have

$$\sigma e_j^{m+1} - \mu (e_j^{m+1})_{ss} = \sum_{l=1}^{m} (M_{m-l} - M_{m-l+1})\delta_t e_j^l + \sigma e_j^m + R_j^{m+1}. \tag{29}$$

Define the functions

$$e^m(s) = \begin{cases} e_j^m, \ s_j - \dfrac{h}{2} < s \leq s_j + \dfrac{h}{2}, \quad 1 \leq j \leq M-1, \\ 0, \ 0 \leq s \leq \dfrac{h}{2} \text{ or } L - \dfrac{h}{2} < s \leq L. \end{cases}$$

and

$$R^m(s) = \begin{cases} R_j^m,\ s_j - \dfrac{h}{2} < s \le s_j + \dfrac{h}{2},\quad 1 \le j \le M-1, \\ 0,\ 0 \le s \le \dfrac{h}{2}\ or\ L - \dfrac{h}{2} < s \le L. \end{cases}$$

We expand the above functions into Fourier series expansions as

$$\begin{cases} e^m(s) = \sum\limits_{l=-\infty}^{\infty} \xi_m(l) \exp^{\frac{I2\pi ls}{L}}, \\ R^m(s) = \sum\limits_{l=-\infty}^{\infty} \lambda_m(l) \exp^{\frac{I2\pi ls}{L}}. \end{cases}$$

where,

$$\begin{cases} \xi_m(l) = \dfrac{1}{L}\displaystyle\int_0^L e^m(s) \exp^{\frac{-I2\pi ls}{L}}\, ds, \\ \lambda_m(l) = \dfrac{1}{L}\displaystyle\int_0^L R^m(s) \exp^{\frac{-I2\pi ls}{L}}\, ds, \end{cases}$$

Applying Parseval's equalities,

$$\int_0^L \|e^m(s)\|^2 ds = \sum_{j=1}^{M-1} h\|e_j^m\|^2, \text{ and}$$
$$\int_0^L \|R^m(s)\|^2 ds = \sum_{j=1}^{M-1} h\|R_j^m\|^2$$

to the above expression, we have

$$\begin{cases} \|e^m\|_2^2 & = \sum_{l=-\infty}^{\infty} \|\xi_m(l)\|^2, \\ \|R^m\|_2^2 & = \sum_{l=-\infty}^{\infty} \|\lambda_m(l)\|^2. \end{cases} \tag{30}$$

Now, we suppose that

$$\begin{cases} e_j^m = \xi_m \exp^{I\sigma_s jh}, \\ R_j^m = \lambda_m \exp^{I\sigma_s jh}, \end{cases} \tag{31}$$

where $\sigma_s = \frac{2\pi l}{L}$. Substituting relations (31) in Equation (29).

$$\begin{aligned} &\xi_{m+1}[\sigma(a_1 \exp^{I\sigma_s(j-1)h} + a_2 \exp^{I\sigma_s(j)h} + a_1 \exp^{I\sigma_s(j+1)h}) - \mu(c_1 \exp^{I\sigma_s(j-1)h} \\ &+ c_2 \exp^{I\sigma_s(j)h} + c_1 \exp^{I\sigma_s(j+1)h})] = \sum_{l=1}^{m} (M_{m-l} - M_{m-l+1})\delta_t \xi_l [a_1 \exp^{I\sigma_s(j-1)h} \\ &+ a_2 \exp^{I\sigma_s(j)h} + a_1 \exp^{I\sigma_s(j+1)h}] + \sigma\xi_m[a_1 \exp^{I\sigma_s(j-1)h} + a_2 \exp^{I\sigma_s(j)h} \\ &+ a_1 \exp^{I\sigma_s(j+1)h}] + \lambda_{m+1}[a_1 \exp^{I\sigma_s(j-1)h} + a_2 \exp^{I\sigma_s(j)h} + a_1 \exp^{I\sigma_s(j+1)h}]. \end{aligned}$$

$$\begin{aligned} \Rightarrow\ &\xi_{m+1}[\sigma(a_2 + 2a_1\cos(\sigma_s h)) - \mu(c_2 + 2c_1\cos(\sigma_s h))] \\ &= \sum_{l=1}^{m} (M_{m-l} - M_{m-l+1})\delta_t \xi_l (a_2 + 2a_1\cos(\sigma_s h)) + \sigma\xi_m(a_2 + 2a_1\cos(\sigma_s h)) \\ &+ \lambda_{m+1}(a_2 + 2a_1\cos(\sigma_s h)). \end{aligned}$$

The above expression is further simplified as

$$\xi_{m+1} = \frac{1}{(\sigma-\mu r)}\sum_{l=1}^{m}(M_{m-l} - M_{m-l+1})(\frac{\xi_l - \xi_{l-1}}{\tau}) + \frac{\sigma}{(\sigma-\mu r)}\xi_m + \frac{1}{(\sigma-\mu r)}\lambda_{m+1}, \quad 1 \le m \le M-1. \tag{32}$$

Theorem 3. *Let ξ_m be the solution of (32), then, there is a positive constant C such that*

$$|\xi_m| \le C(1+\tau)^m|\lambda_1|, \quad m = 0, \ldots, N-1.$$

Proof. We use the mathematical induction to prove this claim. For $m = 1$, we have from (32)

$$|\xi_1| \le |\frac{\lambda_1}{\sigma-\mu r}| \le C(1+\tau)|\lambda_1|.$$

Assume that

$$|\xi_m| \le C(1+\tau)^m|\lambda_1|, \quad m = 0, \ldots, N-2.$$

Now by using the convergence of the series on the RHS of (30), we know that there exists a constant C_2 such that

$$|\lambda_m| \le C_2\tau|\lambda_1|, \quad m = 1, \ldots, N-1.$$

From (32), we have

$$\begin{aligned}
|\xi_{m+1}| &\le \frac{C\tau}{\sigma-\mu r}\sum_{l=1}^{m}|\xi_l - \xi_{l-1}| + \frac{\sigma}{\sigma-\mu r}|\xi_m| + |\frac{\lambda_{m+1}}{\sigma-\mu r}| \\
&= \frac{C\tau}{\sigma-\mu r}\sum_{l=1}^{m}|\xi_m - \xi_0| + \frac{\sigma}{\sigma-\mu r}|\xi_m| + |\frac{\lambda_{m+1}}{\sigma-\mu r}| \\
&= C_1\tau(1+\tau)^m|\lambda_1| + C_3\tau(1+\tau)^m|\lambda_1| + C_2\tau|\lambda_1| \\
&\le (1+\tau)^{m+1}C|\lambda_1|.
\end{aligned}$$

□

Theorem 4. *The scheme (12) is convergent, and the order of convergence is $O(\tau^2 + h^2)$.*

Proof. By Theorem 3, Equation (30) and $m\tau \le T$, we have

$$\begin{aligned}
\|e^m\|_{l_2}^2 &= \sum_{l=-\infty}^{\infty}\|\xi_m(l)\|^2 \le \sum_{l=-\infty}^{\infty}C^2(1+\tau)^{2m}\|\lambda_1(l)\|^2 \\
&= C^2(1+\tau)^{2m}\|R^1\|_{l_2}^2 \\
&\le C^2C_1^2e^{2m\tau}(\tau^2+h^2)^2 \\
&\le C'^2(\tau^2+h^2)^2.
\end{aligned}$$

This completes the proof. □

4. Numerical Findings and Discussion

The efficiency and the validity of the suggested methodologies are confirmed in this part using various test problems by utilizing the L_2 and L_∞ error norms. The numerical results obtained by the proposed schemes are compared. Mathematica 12 was used to obtain the numerical and graphical results.

Example 1. *Consider the time fractional Cattaneo equation,*

$$\frac{\partial v(s,t)}{\partial t} + {}_0^{CF}\mathfrak{D}_t^{\alpha} v(s,t) = \frac{\partial^2 v(s,t)}{\partial^2 s} + g(s,t),\ 1 < \alpha < 2,$$

with initial constraint,

$$v(s,0) = 0,\ \ v_t(s,0) = 0, s > 0,$$

and with boundary constraint,

$$v(0,t) = 0,\ \ v(1,t) = 0,\quad 0 \leqslant t \leqslant 1.$$

The corresponding source term is

$$g(s,t) = 2(1-s^2)s^{\frac{16}{3}}[t + \frac{1}{\alpha-1}(1-\exp(\frac{1-\alpha}{2-\alpha}t))] + t^2(\frac{418}{9}s^{\frac{16}{3}} - \frac{208}{9}s^{\frac{10}{3}}).$$

The analytic solution of the given problem is $v(s,t) = t^2(1-s^2)s^{\frac{16}{3}}$. The suggested schemes are implemented on the aforementioned problem to obtain the numerical results. The errors obtained by the schemes are compared with each other in Tables 1–3. Figure 1 presents an efficient comparison of approximate and exact solutions at various times. Figure 2 exhibits the 2D error profile. The 3D comparison between the exact and approximate solutions is depicted in Figure 3. The approximate solution using the scheme based on cubic B-splines when $\tau = 0.01$ and $M = 20$ at $T = 0.5$ and $T = 1$ for Example 1 are given by

$$V(s,0.5) = \begin{cases} -2.3293 \times 10^{-21} + 1.9849 \times 10^{-4}s - 3.03577 \times 10^{-18}s^2 + 1.5661 \times 10^{-3}s^3, & s \in [0, \frac{1}{20}) \\ -8.7474 \times 10^{-7} + 2.5097 \times 10^{-4}s - 1.0497 \times 10^{-3}s^2 + 8.5639 \times 10^{-3}s^3, & s \in [\frac{1}{20}, \frac{1}{10}) \\ -1.7230 \times 10^{-5} + 7.4162 \times 10^{-4}s - 5.9562 \times 10^{-3}s^2 + 0.0249s^3, & s \in [\frac{1}{10}, \frac{3}{20}) \\ \vdots & \\ \vdots & \\ 1.5989 - 6.1887s + 8.0414s^2 - 3.4502s^3, & s \in [\frac{17}{20}, \frac{9}{10}) \\ 2.5132 - 9.2363s + 11.4277s^2 - 4.7044s^3, & s \in [\frac{9}{10}, \frac{19}{20}) \\ 3.8346 - 13.4089s + 15.8199s^2 - 6.2455s^3, & s \in [\frac{19}{20}, 1). \end{cases}$$

and

$$V(s,1) = \begin{cases} -8.3009 \times 10^{-20} + 1.4517 \times 10^{-3}s + 2.9490 \times 10^{-17}s^2 + 6.8016 \times 10^{-3}s^3, & s \in [0, \frac{1}{20}) \\ -3.5058 \times 10^{-6} + 1.6620 \times 10^{-3}s - 4.2069 \times 10^{-3}s^2 + 3.485 \times 10^{-2}s^3, & s \in [\frac{1}{20}, \frac{1}{10}) \\ -6.8991 \times 10^{-5} + 3.6266 \times 10^{-3}s - 2.3853 \times 10^{-2}s^2 + 0.1003s^3, & s \in [\frac{1}{10}, \frac{3}{20}) \\ \vdots & \\ \vdots & \\ 6.3895 - 24.7327s + 32.1435s^2 - 13.7945s^3, & s \in [\frac{17}{20}, \frac{9}{10}) \\ 10.0419 - 36.9074s + 45.6709s^2 - 18.8046s^3, & s \in [\frac{9}{10}, \frac{19}{20}) \\ 15.3213 - 53.5792s + 63.2202s^2 - 24.9623s^3, & s \in [\frac{19}{20}, 1). \end{cases}$$

respectively.

Table 1. Comparison of errors using various B-splines when $\alpha = 1.1$, $dt = 0.001$, $T = 1$ for Example 1.

M	CuBS		TCuBS		ECuBS	
	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm
20	1.63×10^{-3}	2.60×10^{-3}	1.65×10^{-3}	2.64×10^{-3}	9.46×10^{-4}	1.47×10^{-3}
40	4.11×10^{-4}	6.55×10^{-4}	4.16×10^{-4}	6.65×10^{-4}	2.45×10^{-4}	3.94×10^{-4}
80	1.08×10^{-4}	1.71×10^{-4}	1.09×10^{-4}	1.73×10^{-4}	6.61×10^{-5}	1.04×10^{-4}
160	3.20×10^{-5}	5.00×10^{-5}	3.23×10^{-5}	5.06×10^{-5}	2.03×10^{-5}	3.25×10^{-5}

Table 2. Comparison of errors using various B-splines with $\alpha = 1.5$, $dt = 0.001$, $T = 1$ for Example 1.

M	CuBS		TCuBS		ECuBS	
	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm
20	1.54×10^{-3}	2.49×10^{-3}	1.56×10^{-3}	2.53×10^{-3}	1.42×10^{-3}	8.89×10^{-4}
40	3.90×10^{-4}	6.27×10^{-4}	3.94×10^{-4}	6.37×10^{-4}	2.33×10^{-4}	3.77×10^{-4}
80	1.02×10^{-4}	1.64×10^{-4}	1.04×10^{-4}	1.67×10^{-4}	6.42×10^{-5}	9.99×10^{-5}
160	3.08×10^{-5}	4.83×10^{-5}	3.11×10^{-5}	4.89×10^{-5}	2.00×10^{-5}	3.21×10^{-5}

Table 3. Comparison of errors using various B-splines with $\alpha = 1.9$, $dt = 0.001$, $T = 1$ for Example 1.

M	CuBS		TCuBS		ECuBS	
	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm
20	1.39×10^{-3}	2.31×10^{-3}	1.41×10^{-3}	2.35×10^{-3}	8.01×10^{-4}	1.30×10^{-3}
40	3.50×10^{-4}	5.85×10^{-4}	3.56×10^{-4}	5.95×10^{-4}	2.09×10^{-4}	3.44×10^{-4}
80	9.18×10^{-5}	1.51×10^{-4}	9.30×10^{-5}	1.54×10^{-4}	5.70×10^{-5}	9.18×10^{-5}
160	2.73×10^{-5}	4.29×10^{-5}	2.75×10^{-5}	4.35×10^{-5}	1.85×10^{-5}	2.96×10^{-5}

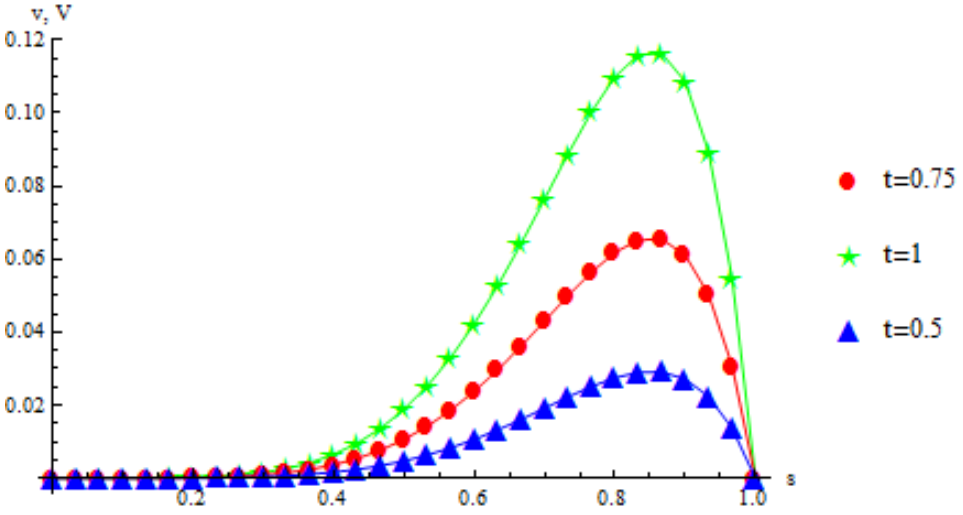

Figure 1. The exact and approximate (triangles, stars, circles) solutions using cubic B-spline-based scheme for Example 1 at various times when $h = \frac{1}{60}$.

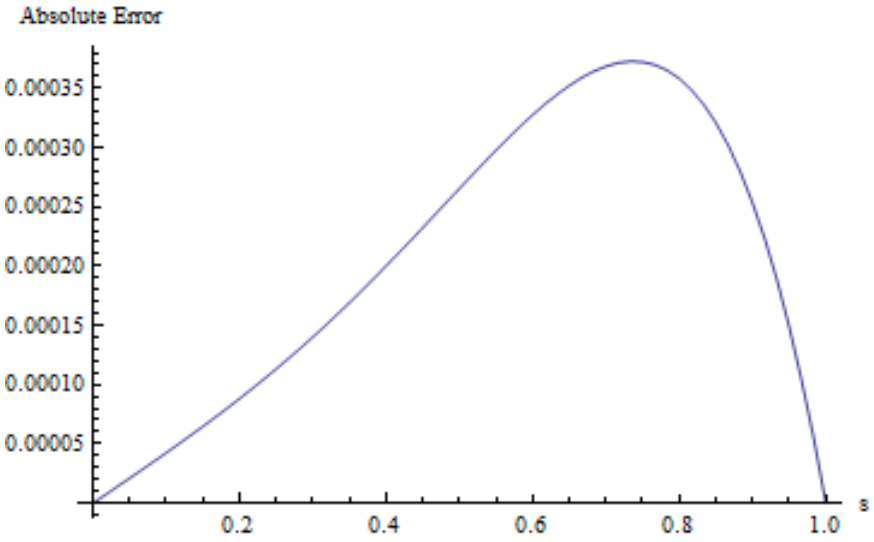

Figure 2. The 2D error profile using cubic B-spline-based scheme for Example 1 when $h = \frac{1}{60}$, $T = 1$, $dt = 0.01$, $\alpha = 1.5$.

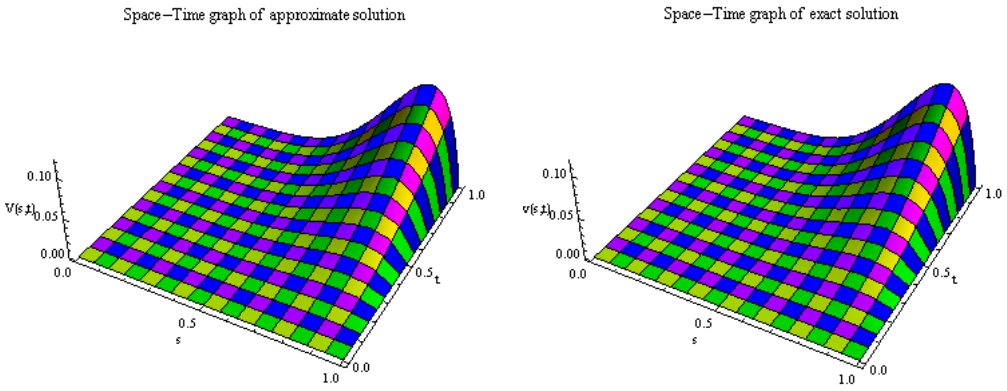

Figure 3. The approximate (**left**) and exact (**right**) solutions using cubic B-spline-based scheme for Example 1 when $h = \frac{1}{60}$, $T = 1$, $dt = 0.01$, $\alpha = 1.5$.

Example 2. *Consider the time fractional Cattaneo equation,*

$$\frac{\partial v(s,t)}{\partial t} + {}_0^{CF}\mathfrak{D}_t^{\alpha} v(s,t) = \frac{\partial^2 v(s,t)}{\partial^2 s} + g(s,t),\ 1 < \alpha < 2,$$

with ICs,

$$v(s,0) = 0,\ \ v_t(s,0) = \sin x,\ \ 0 \leqslant s \leqslant 1,$$

and BCs,

$$v(0,t) = 0,\ \ v(1,t) = t\sin(1),\ \ t > 0.$$

The corresponding source term is $g(s,t) = (1+t)\sin s$. The analytic solution of the given problem is $v(s,t) = t\sin s$. In order to achieve the desired numerical results the presented schemes are applied on Example 2. The errors obtained by the schemes are compared with each other in Tables 4–6. For various time stages, a sharp contrast between the exact and approximate solutions is presented in Figure 4. The 2D absolute error profile is plotted in Figure 5. Figure 6 depicts a 3D comparison between the exact and approximate solutions.

The approximate solution using cubic B-spline-based scheme when $\tau = 0.01$ and $M = 20$ at $T = 0.5$ and $T = 1$ for Example 2 are given by

$$V(s,0.5)=\begin{cases}3.0358\times10^{-17}+0.499995s+4.4409\times10^{-15}s^2-0.08331s^3, & s\in[0,\frac{1}{20})\\ -2.6046\times10^{-8}+0.499996s-3.12551\times10^{-5}s^2-0.08311s^3, & s\in[\frac{1}{20},\frac{1}{10})\\ -4.4227\times10^{-7}+0.50001s-1.56124\times10^{-4}s^2-0.08267s^3, & s\in[\frac{1}{10},\frac{3}{20})\\ \vdots & \\ \vdots & \\ -7.78029\times10^{-3}+0.5336s-0.05168s^2-0.05339s^3, & s\in[\frac{17}{20},\frac{9}{10})\\ -0.01016+0.5415s-0.06050s^2-0.05013s^3, & s\in[\frac{9}{10},\frac{19}{20})\\ -0.01307+0.5507s-0.07016s^2-0.04673s^3, & s\in[\frac{19}{20},1).\end{cases}$$

and

$$V(s,1)=\begin{cases}-4.5103\times10^{-17}+0.99997s-5.32907\times10^{-14}s^2-0.1666s^3, & s\in[0,\frac{1}{20})\\ -5.2096\times10^{-8}+0.99998s-6.25146\times10^{-5}s^2-0.1662s^3, & s\in[\frac{1}{20},\frac{1}{10})\\ -8.8457\times10^{-7}+s-3.12256\times10^{-4}s^2-0.1654s^3, & s\in[\frac{1}{10},\frac{3}{20})\\ \vdots & \\ \vdots & \\ -1.5548\times10^{-2}+1.0671s-0.1033s^2-0.1068s^3, & s\in[\frac{17}{20},\frac{9}{10})\\ -2.0307\times10^{-2}+1.08297s-0.1209s^2-0.1003s^3, & s\in[\frac{9}{10},\frac{19}{20})\\ -2.6118\times10^{-2}+1.10132s-0.1402s^2-0.0935s^3, & s\in[\frac{19}{20},1).\end{cases}$$

respectively.

Table 4. Comparison of errors using various B-splines with $\alpha=1.1$, $dt=0.001$, $T=1$ for Example 2.

M	CuBS		TCuBS		ECuBS	
	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm
20	7.15×10^{-6}	9.97×10^{-6}	6.71×10^{-6}	9.35×10^{-6}	2.87×10^{-7}	3.99×10^{-7}
40	1.79×10^{-6}	2.49×10^{-6}	1.68×10^{-6}	2.34×10^{-6}	1.43×10^{-8}	1.98×10^{-8}
80	4.47×10^{-7}	6.23×10^{-7}	4.19×10^{-7}	5.84×10^{-7}	6.15×10^{-9}	8.57×10^{-9}
160	1.12×10^{-7}	1.56×10^{-7}	1.05×10^{-7}	1.46×10^{-7}	1.79×10^{-10}	2.50×10^{-10}

Table 5. Comparison of errors using various B-splines with $\alpha=1.5$, $dt=0.001$, $T=1$ for Example 2.

M	CuBS		TCuBS		ECuBS	
	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm
20	7.04×10^{-6}	9.81×10^{-5}	6.60×10^{-6}	9.20×10^{-6}	2.82×10^{-7}	3.93×10^{-7}
40	1.76×10^{-6}	2.45×10^{-6}	1.65×10^{-6}	2.30×10^{-6}	1.40×10^{-8}	1.96×10^{-8}
80	4.40×10^{-7}	6.13×10^{-7}	4.12×10^{-7}	5.75×10^{-7}	6.05×10^{-9}	8.44×10^{-9}
160	1.09×10^{-7}	1.53×10^{-7}	1.03×10^{-7}	1.44×10^{-7}	1.76×10^{-10}	2.46×10^{-10}

Table 6. Comparison of errors using various B-splines with $\alpha=1.9$, $dt=0.001$, $T=1$ for Example 2.

M	CuBS		TCuBS		ECuBS	
	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm
20	7.20×10^{-6}	1.01×10^{-5}	6.75×10^{-6}	9.43×10^{-6}	2.89×10^{-7}	4.03×10^{-7}
40	1.80×10^{-6}	2.51×10^{-6}	1.69×10^{-6}	2.36×10^{-6}	1.44×10^{-8}	2.01×10^{-8}
80	4.50×10^{-7}	6.29×10^{-7}	4.22×10^{-7}	5.90×10^{-7}	6.19×10^{-9}	8.65×10^{-9}
160	1.12×10^{-7}	1.57×10^{-7}	1.05×10^{-7}	1.47×10^{-7}	1.80×10^{-10}	2.52×10^{-10}

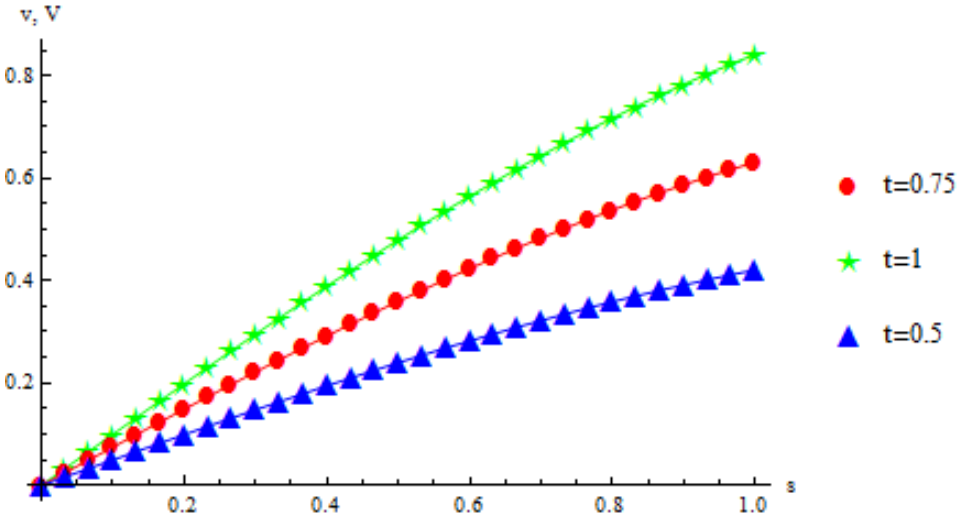

Figure 4. The exact and approximate (triangles, stars, circles) solutions using cubic B-spline-based scheme for Example 2 at various times when $h = \frac{1}{60}$.

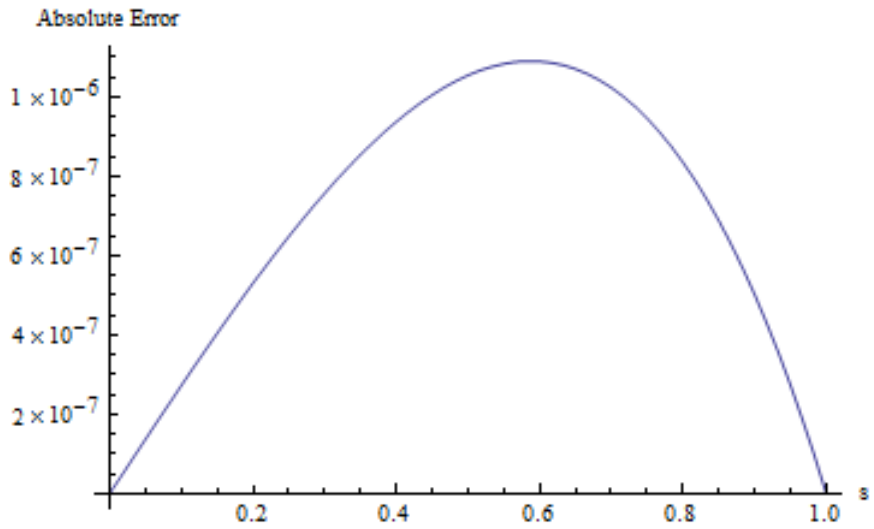

Figure 5. The 2D error profile using cubic B-spline-based scheme for Example 2 when $h = \frac{1}{60}$, $T = 1$, $dt = 0.01$, $\alpha = 1.5$.

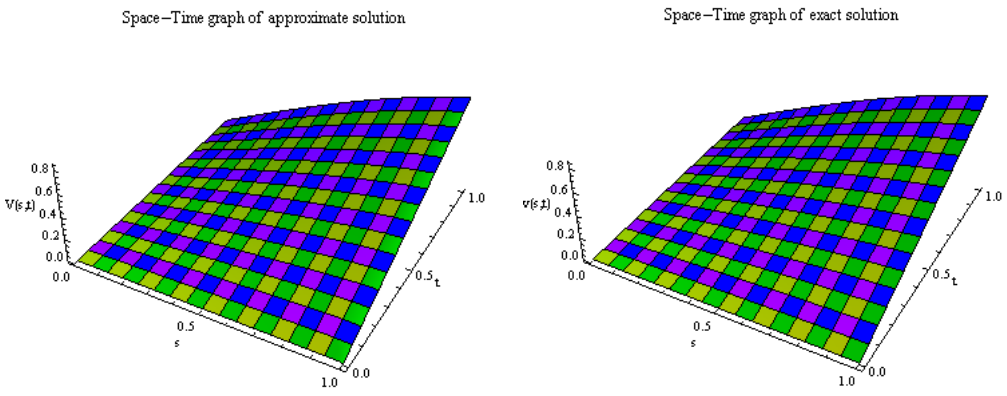

Figure 6. The approximate (**left**) and exact (**right**) solutions using cubic B-spline-based scheme for Example 2 when $h = \frac{1}{60}$, $T = 1$, $dt = 0.01$, $\alpha = 1.5$.

Example 3. *Consider the time fractional Cattaneo equation,*

$$\frac{\partial v(s,t)}{\partial t} + {}_0^{CF}\mathfrak{D}_t^{\alpha} v(s,t) = \frac{\partial^2 v(s,t)}{\partial^2 s} + g(s,t),\ 1 < \alpha < 2,$$

with ICs,

$$v(s,0) = 0,\ \ v_t(s,0) = (1-s)\cos s,\ \ 0 \leqslant s \leqslant 1,$$

and BCs

$$v(0,t) = t,\ \ v(1,t) = 0,\ \ t > 0.$$

The corresponding source term is $g(s,t) = (1+t)(1-s)\cos s - 2t\sin s$. The analytic solution of the given problem is $v(s,t) = t(1-s)\cos s$. The proposed methodologies are utilized to acquire the numerical results for Example 3.

A comparison of computed errors is provided in Tables 7–9. For various time stages, a close comparison between the exact and approximate solutions is displayed in Figure 7. The 2D error function is plotted in Figure 8. Figure 9 depicts a 3D comparison between the exact and approximate solutions.

The approximate solution using cubic B-spline-based scheme when $\tau = 0.01$ and $M = 20$ at $T = 0.5$ and $T = 1$ for Example 3 are given by

$$V(s,0.5) = \begin{cases} 0.5 - 0.5s - 0.25s^2 + 0.2519s^3, & s \in [0, \frac{1}{20}) \\ 0.5 - 0.49997s - 0.2505s^2 + 0.2551s^3, & s \in [\frac{1}{20}, \frac{1}{10}) \\ 0.499998 - 0.4999s - 0.2511s^2 + 0.2571s^3, & s \in [\frac{1}{10}, \frac{3}{20}) \\ \vdots & \\ \vdots & \\ 0.5282 - 0.6185s - 0.07785s^2 + 0.1681s^3, & s \in [\frac{17}{20}, \frac{9}{10}) \\ 0.5375 - 0.6496s - 0.04329s^2 + 0.1553s^3, & s \in [\frac{9}{10}, \frac{19}{20}) \\ 0.5491 - 0.6860s - 4.990 \times 10^{-3}s^2 + 0.1419s^3, & s \in [\frac{19}{20}, 1). \end{cases}$$

and

$$V(s,1) = \begin{cases} 1 - 0.9999s - 0.5s^2 + 0.5039s^3, & s \in [0, \frac{1}{20}) \\ 0.9999 - 0.9999s - 0.5009s^2 + 0.5101s^3, & s \in [\frac{1}{20}, \frac{1}{10}) \\ 0.9999 - 0.9998s - 0.5022s^2 + 0.5142s^3, & s \in [\frac{1}{10}, \frac{3}{20}) \\ \vdots & \\ \vdots & \\ 1.0563 - 1.2367s - 0.1560s^2 + 0.3364s^3, & s \in [\frac{17}{20}, \frac{9}{10}) \\ 1.0750 - 1.2989s - 0.0869s^2 + 0.3108s^3, & s \in [\frac{9}{10}, \frac{19}{20}) \\ 1.0980 - 1.3716s - 0.0103s^2 + 0.2839s^3, & s \in [\frac{19}{20}, 1). \end{cases}$$

respectively.

Table 7. Comparison of errors using various B-splines with $\alpha = 1.1,\ dt = 0.001,\ T = 1$ for Example 3.

M	CuBS		TCuBS		ECuBS	
	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm
20	2.21×10^{-5}	3.19×10^{-5}	2.23×10^{-6}	3.55×10^{-6}	4.04×10^{-6}	1.47×10^{-6}
40	5.53×10^{-6}	7.99×10^{-6}	5.58×10^{-7}	8.89×10^{-7}	1.01×10^{-6}	3.94×10^{-6}
80	1.38×10^{-6}	1.99×10^{-6}	1.40×10^{-7}	2.23×10^{-7}	2.97×10^{-7}	1.04×10^{-7}
160	3.46×10^{-7}	4.99×10^{-7}	3.49×10^{-8}	5.57×10^{-8}	6.71×10^{-8}	3.25×10^{-8}

Table 8. Comparison of errors using various B-splines with $\alpha = 1.5,\ dt = 0.001,\ T = 1$ for Example 3.

M	CuBS		TCuBS		ECuBS	
	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm
20	2.18×10^{-5}	3.14×10^{-5}	2.24×10^{-6}	3.56×10^{-6}	4.06×10^{-6}	6.07×10^{-6}
40	5.44×10^{-6}	7.88×10^{-6}	5.60×10^{-7}	8.90×10^{-7}	1.01×10^{-6}	1.51×10^{-6}
80	1.36×10^{-6}	1.97×10^{-6}	1.40×10^{-7}	2.23×10^{-7}	2.98×10^{-7}	4.58×10^{-7}
160	3.40×10^{-7}	4.92×10^{-7}	3.50×10^{-8}	5.57×10^{-8}	6.74×10^{-8}	9.56×10^{-8}

Table 9. Comparison of errors using various B-splines with $\alpha = 1.9,\ dt = 0.001,\ T = 1$ for Example 3.

M	CuBS		TCuBS		ECuBS	
	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm	L_2 Norm	L_∞ Norm
20	2.23×10^{-5}	3.23×10^{-5}	2.33×10^{-6}	3.69×10^{-6}	4.22×10^{-6}	6.30×10^{-6}
40	5.57×10^{-6}	8.08×10^{-6}	5.81×10^{-7}	9.24×10^{-7}	1.05×10^{-6}	1.57×10^{-6}
80	1.39×10^{-6}	2.02×10^{-6}	1.45×10^{-7}	2.32×10^{-7}	3.09×10^{-7}	4.76×10^{-7}
160	3.48×10^{-7}	5.05×10^{-7}	3.63×10^{-8}	5.79×10^{-8}	7.20×10^{-8}	9.96×10^{-8}

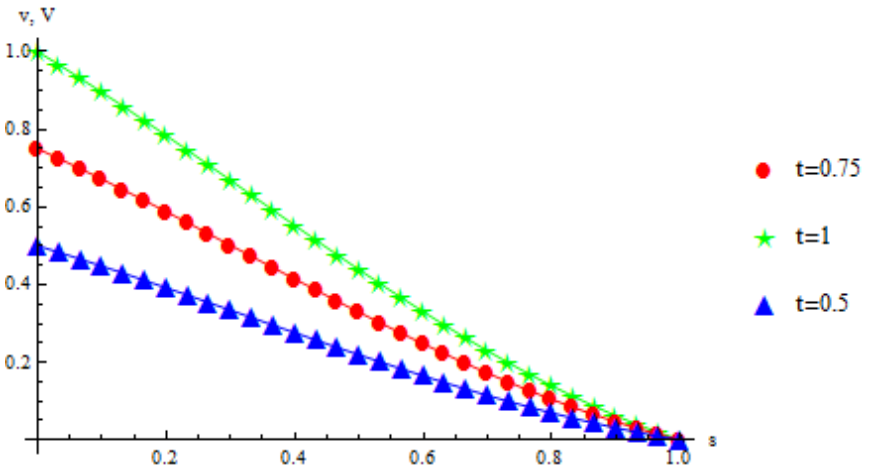

Figure 7. The exact and approximate (triangles, stars, circles) solutions using cubic B-spline-based scheme for Example 3 at various times when $h = \frac{1}{60}$.

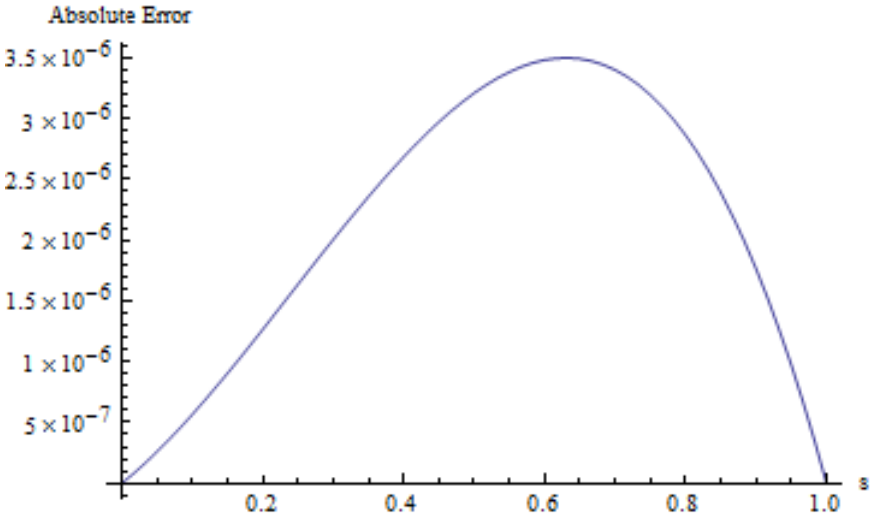

Figure 8. The 2D error profile using cubic B-spline-based scheme when for Example 2 when $h = \frac{1}{60}$, $T = 1$, $dt = 0.01$, $\alpha = 1.5$.

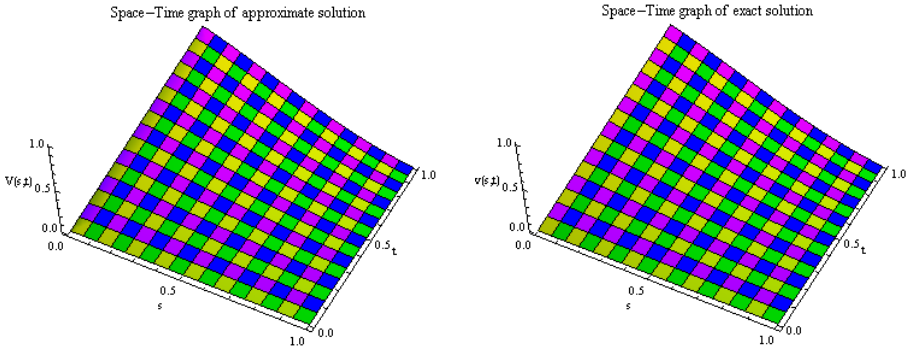

Figure 9. The approximate (**left**) and exact (**right**) solutions using cubic B-spline-based scheme for Example 3 when $h = \frac{1}{60}$, $T = 1$, $dt = 0.01$, $\alpha = 1.5$.

5. Concluding Remarks

The spline-based collocation schemes are developed for the numerical solution of the time fractional Cattaneo differential equation involving the Caputo–Fabrizio time fractional derivative. To begin with, the space derivative involved is approximated using the cubic B-spline. Secondly, using finite differences, the Caputo–Fabrizio derivative is approximated. The stability and convergence analysis of the schemes are also discussed in detail. The splines used are the cubic B-splines, extended cubic B-splines and the trigonometric cubic B-splines. The key advantage is that the approximate solution is obtained as a piecewise continuous function so that approximate solution at any desired position in the domain can be tracked. The efficiency and accuracy of the proposed approaches are confirmed by the experimental findings. The suggested schemes can be applied to a wide range of problems in varied fields of applied sciences.

Author Contributions: Conceptualization, M.Y. , Q.U.N.A. and S.K.; methodology, Q.U.N.A. and R.G.; validation, R.G. and S.K.; formal analysis, M.Y., Q.U.N.A. and S.K.; investigation, S.K.; writing—original draft preparation, Q.U.N.A.; writing—review and editing, M.Y., R.G. and S.K.; supervision, R.G. and M.Y.; project administration, R.G.; funding acquisition, R.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by Deanship of Scientific Research, Prince Sattam bin Abdulaziz University, Alkharj, Saudi Arabia.

Institutional Review Board Statement: Not applicable:

Informed Consent Statement: Not applicable.

Data Availability Statement: No data were used.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, Z.; Cheng, A.; Li, X. A second order crank-Nicolson scheme for fractional Cattaneo equation based on new fractional derivative. *Appl. Math. Comput.* **2017**, *311*, 361–374. [CrossRef]
2. Zheng, B. (G'/G)-expansion method for solving fractional partial differential equations in the theory of mathematical physics. *Commun. Theor. Phys.* **2012**, *58*, 623–630. [CrossRef]
3. Tavares, D.; Almeida, R.; Torres, D.F.M. Caputo derivatives of fractional variable order: Numerical approximations. *Commun. Nonlinear Sci. Numer. Simul.* **2016**, *35*, 69–87. [CrossRef]
4. Baleanu, D.; Rezapour, S.; Mohammadi, H. Some existence results on nonlinear fractional differential equations. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2013**, *371*, 20120144. [CrossRef] [PubMed]
5. Diethelm, K.; Ford, N.J.; Freed, A.D. A predictor-corrector approach for the numerical solution of fractional differential equations. *Nonlinear Dyn.* **2002**, *29*, 3–22. [CrossRef]
6. Meerschaert, M.M.; Tadjeran, C. Finite difference approximations for fractional advection–dispersion flow equations. *J. Comput. Appl. Math.* **2004**, *172*, 65–77. [CrossRef]
7. Hashim, I.; Abdulaziz, O.; Momani, S. Homotopy analysis method for fractional IVPS. *Commun. Nonlinear Sci. Numer. Simul.* **2009**, *14*, 674–684. [CrossRef]

8. Suarez, L.; Shokooh, A. An eigenvector expansion method for the solution of motion containing fractional derivatives. *J. Appl. Mech.* **1997**, *64*, 629–635. [CrossRef]
9. Jafari, H.; Yousefi, S.; Firoozjaee, M.; Momani, S.; Khalique, C.M. Application of Legendre wavelets for solving fractional differential equations. *Comput. Math. Appl.* **2011**, *62*, 1038–1045. [CrossRef]
10. Li, Y.; Zhao, W. Haar wavelet operational matrix of fractional order integration and its applications in solving the fractional order differential equations. *Appl. Math. Comput.* **2010**, *216*, 2276–2285. [CrossRef]
11. Yuanlu, L. Solving a nonlinear fractional differential equation using Chebyshev wavelets. *Commun. Nonlinear Sci. Numer. Simul.* **2010**, *15*, 2284–2292.
12. Li, Y.; Sun, N. Numerical solution of fractional differential equations using the generalized block pulse operational matrix. *Comput. Math. Appl.* **2011**, *62*, 1046–1054. [CrossRef]
13. Odibat, Z. On Legendre polynomial approximation with the vim or ham for numerical treatment of nonlinear fractional differential eqations. *J. Comput. Appl. Math.* **2011**, *235*, 2956–2968. [CrossRef]
14. Araci, S. Novel identities for q-Genocchi numbers and polynomials. *J. Funct. Spaces Appl.* **2012**, *2012*, 214961. [CrossRef]
15. Gürbüz, B.; Sezer, M. Laguerre polynomial solutions of a class of initial and boundary value problems arising in science and engineering fields. *Acta Phys. Pol. A* **2016**, *130*, 194–197. [CrossRef]
16. Caputo, M.; Fabrizio, M. A new definition of fractional derivative without singular kernel. *Prog. Fract. Differ. Appl.* **2015**, *1*, 73–85.
17. Caputo, M.; Fabrizio, M. Applications of new time and spatial fractional derivatives with exponential kernels. *Prog. Fract. Differ. Appl.* **2016**, *2*, 1–11. [CrossRef]
18. Feulefack, P.A.; Djida, J.D.; Abdon, A. A new model of groundwater flow within an unconfined aquifer: Application of Caputo-Fabrizio fractional derivative. *Discret. Cont. Dyn. Syst.-B* **2019**, *24*, 3227–3247. [CrossRef]
19. Delgado, V.F.M.; Aguilar, J.F.G.; Saad, K.; Jiménez, R.F.E. Application of the Caputo-Fabrizio and Atangana-Baleanu fractional derivatives to mathematical model of cancer chemotherapy effect. *Math. Methods Appl. Sci.* **2019**, *42*, 1167–1193. [CrossRef]
20. Abdon, A. On the new fractional derivative and application to nonlinear fishers reaction-diffusion equation. *Appl. Math. Comput.* **2016**, *273*, 948–956.
21. Liu, Z.; Cheng, A.; Li, X. A fully discrete spectral method for fractional Cattaneo equation based on Caputo-Fabrizo derivative. *Numer. Methods Partial Differ. Equ.* **2019**, *35*, 936–954. [CrossRef]
22. Compte, A.; Metzler, R. The generalized Cattaneo equation for the description of anomalous transport processes. *J. Phys. A* **1997**, *30*, 72–77. [CrossRef]
23. Giusti, A. Why fractional derivatives with nonsingular kernels should not be used. *Nonlinear Dyn.* **2018**, *93*, 1757–1763. [CrossRef]
24. Li, X.; Rui, H.; Liu, Z. A block-centered finite difference method for fractional Cattaneo equaton. *Numer. Methods Partial Differ. Equ.* **2018**, *34*, 296–316. [CrossRef]
25. Dhiman, N.; Huntul, M.J.; Tamsir, M. A modified trigonometric cubic B-spline collocation technique for solving the time-fractional diffusion equation. *Eng. Comput.* **2021**, *38*. [CrossRef]
26. Tamsir, M.; Dhiman, N.; Gill, F.C. Approximation of 3D convection diffusion equation using DQM based on modified cubic trigonometric B-splines. *J. Comput. Methods Sci. Eng.* **2020**, *20*, 1357–1366.
27. Tamsir, M.; Dhiman, N. DQM Based on the Modified Form of CTB Shape Functions for Coupled Burgers' Equation in 2D and 3D. *Int. J. Math. Eng. Manag. Sci.* **2019**, *4*, 1051–1067.
28. Dhiman, N.; Tamsir, M. A collocation technique based on modified form of trigonometric cubic B-spline basis functions for Fisher's reaction-diffusion equation. *Multidiscip. Model. Mater. Struct.* **2018**, *14*, 923–939. [CrossRef]
29. Tamsir, M. Cubic trigonometric B-spline differential quadrature method for numerical treatment of Fisher's reaction-diffusion equations. *Alex. Eng. J.* **2018**, *53*, 2019–2026.
30. Abbas, M.; Majid, A.A.; Ismail, A.I.M.; Rashid, A. Numerical method using cubic b-spline for a strongly coupled reaction-diffusion system. *PLoS ONE* **2014**, *9*, e83265.
31. Abbas, M.; Majid, A.A.; Ismail, A.I.M.; Rashid, A. The application of cubic trigonometric B-spline to the numerical solution of the hyperbolic problems. *Appl. Math. Comput.* **2014**, *239*, 74–88. [CrossRef]
32. Morton, K.W.; Mayers, D.F. *Numerical Solution of Partial Differential Equations: An Introduction*; Cambridge University Press: Cambridge, UK, 2005.
33. Smith, G.D. *Numerical Solution of Partial Differential Equations: Finite Difference Methods*; Oxford University Press: Oxford, UK, 1987.