Paradigm International Journal of Mathematics and Computer in Engineering 4(1) (2026) 211–234

International Journal of Mathematics and Computer in Engineering

https://reference-global.com/journal/IJMCE

Original Study

An efficient higher-order trigonometric cubic B-spline collocation method for time-fractional Burgers equations

Murat Önal[1], Berat Karaagac[2] †, Alaattin Esen[1]

[1]İnönü University, Faculty of Science and Arts, Department of Mathematics, Malatya, 44280, Türkiye

[2]Tarsus University, Faculty of Engineering, Department of Natural and Mathematical Sciences, Mersin, 33400, Türkiye

Communicated by Haci Mehmet Baskonus; Received: 07.01.2026; Accepted: 12.02.2026; Online: 18.03.2026

Abstract

This manuscript is devoted to investigate the numerical solutions of the nonlinear time-fractional Burgers equation representing a significant extension of the classical Burgers equation to fractional derivative. For this purpose, an efficient higher-order trigonometric cubic B-spline collocation method, which is based on finite element analysis, is presented and used to achieve the aim of this work. While obtaining the numerical solutions of the mentioned equation, the discretization of the spatial part is performed via the Crank-Nicolson approach and the time derivative is performed in Caputo sense and the discretization of the time derivative is made by $L1$ algorithm. Also, the nonlinear term seen in the Burgers equation is linearized through the use of the Rubin-Graves linearization technique. Consequently, the performing of the collocation method is resulted to obtain a numerical scheme which is producing an algebraic system being solved by iteratively. The stability of the numerical scheme is investigated using the von-Neumann stability criteria. Three test problems are considered to confirm the validity, accuracy and efficiency of the method. The error between numerical solutions and exact ones is measured with the norms L_2 and L_∞. Comparison results are presented by tables, the behaviour of the numerical solutions and the harmony with the exact solutions are depicted with graphs as well.

Keywords: Time-fractional Burgers equation, higher-order method, trigonometric cubic B-spline, collocation method, stability.
AMS 2020 codes: 65L60; 65M60; 65M12; 35R11; 26A33.

1 Introduction

The history of fractional analysis, which is a generalized form of the derivative and integral topics discussed in classical analysis to non-fractional order derivatives and integrals, dates back to the 1695 notes between Leibniz and L'hospital discussing the $(1/2)^{th}$ order derivative of a function. Despite the early interest of mathematicians such as Euler, Laplace and Fourier, fractional analysis remained a purely theoretical field for more than two centuries. However, during the last few centuries many researchers have shown that fractional order analysis is more adequate than classical analysis in describing many real materials. The most fundamental and important property of fractional order operators is their ability to describe the memory hereditary effects of many materials and physical processes [1]. For example, in health science the resistance of the brain to drug on tumor cells and stages of a tumor can be described by fractional calculus [2]. Similarly, in

†Corresponding author.
Email address: beratkaraagac@tarsus.edu.tr

Paradigm
ISSN 2956-7068 doi:10.2478/ijmce-2026-0013

the study of deforestation and its effects on the biodiversity of wildlife species, fractional order differential equations are used due to their ability to deal with complex systems [3]. Over the years, a large number of mathematicians have presented various definitions that fit the idea of a non-integer integral or derivative, using their own notations and approaches. Grünwald and Letnikov introduced a limit-based formulation in the 1860s, now known as the Grünwald–Letnikov fractional derivative [4]. Shortly thereafter, Riemann formalized what became known as the Riemann-Liouville derivative [1]. Since then, many applications of the fractional derivatives and integrals of the Riemann–Liouville type have been demonstrated in various domains of science and technology. In the early twentieth century, Caputo provided an alternative definition of a fractional derivative related to the Riemann–Liouville integral, which gained popularity due to its compatibility with classical initial conditions [5]. The Riemann–Liouville derivative has its foundation in theory, though the Caputo derivative is frequently used in physical modeling due to its suitability with conventional initial and boundary conditions. In addition to the definitions presented, the concept of fractional analysis has evolved over the centuries due to the industrialization of the complexities associated with a heterogeneous formation, the ability of fractional operators to capture the behavior of multi-dimensional environments due to their diffusion operations, and the rapid development of mathematical techniques using computer software, allowing researchers to present their perspectives while analyzing complex formations, providing new definitions and theorems, hundreds of books, and scientific publications. Also, significant progress has been made in the numerical analysis of nonlinear fractional differential equations, including studies on fractional Riccati equations, complex spatiotemporal dynamics in fractional systems, fractional ecological models, and fractional Burger-Huxley equations, highlighting the growing interest in accurate and efficient numerical schemes for nonlinear fractional problems [6–9]. In this study, we are going to consider the fractional Burgers equation with the boundary and initial conditions given as follows, respectively;

$$D_t^{\gamma} u + u u_x - \nu u_{xx} = f(x,t), \tag{1}$$

and,

$$\begin{aligned} &u(a,t) = h_1(t), \;\; u(b,t) = h_2(t), \\ &u_x(a,t) = h_3(t), \, u_x(b,t) = h_4(t), \\ &u(x,0) = g(x), \end{aligned} \tag{2}$$

where $x \in [a,b], t \in [0,T]$ and ν is the viscosity parameter, D^{γ} symbolize the fractional order derivative. The fractional Burgers equation, which belongs to the class of fractional order partial differential equations, has a wide range of applications in various fields such as heat transfer processes, dispersed water phenomena, sound and shallow water wave propagation analysis. Due to its wide range of applications in modeling nonlinear and anomalous diffusion phenomena, the fractional Burgers equation has garnered significant attention from researchers. Various numerical methods have been developed to solve the equation, as evidenced by numerous studies in literature. For instance, Esen and Tasbozan [10] have obtained the numerical solutions of the fractional order Burgers equation using the cubic B-spline Collocation method. Singh and Gupta [11], in their study which is based on the trigonometric tension B-spline collocation method, have addressed the fractional-order Burgers equation. Duangpan *et al.* [12] have obtained results for the fractional Burgers equation by a finite integration approach based on Chebyshev polynomials. In [13], Tamsir *et al.* have applied Hybrid B-spline collocation approach to obtain numerical results of the equation. And, Jeon and Bu [14] have obtained solutions of the Burgers equation using Adams–Moulton and linearized technique. A direct numerical method for solving time-fractional Burger's equation using a modified hybrid B-spline basis function is presented by Hashmi *et al.* in [15]. In [16], Sadri *et al.* developed a novel pseudo-operational collocation method using bivariate Jacobi polynomials and investigated parameter effects, error bounds, and existence-uniqueness results for the time-fractional Burgers equation. In [17], Yaseen and Abbas introduced a cubic trigonometric B-spline method combined with a linearization technique to efficiently solve the time-fractional Burgers equation and established unconditional stability. In [18], Al-saedi and Rashindinia have applied the Galerkin Finite element method based on cubic B-spline basis function to the fractional-order

Burgers equation. Maji and Natesan [19] have investigated the numerical results of the equation using the discontinuous Galerkin method. Raul and Rohil [20] have obtained numerical solutions of the problem with the optimal quantic B-spline method. Additionally, Taghipour and Aminikhah have used the Pell collocation method for numerical results in the paper referred as [21]. Danaf and Hadhoud [22] have used parametric spline functions and obtained numerical solutions.

In this paper, we are going to consider the fractional Burgers equation using efficient higher-order collocation approach based on trigonometric cubic B-splines. Although various spline-based collocation methods have been proposed for the time-fractional Burgers equation, most existing approaches are limited to the second- or third-order spatial accuracy, or rely on polynomial B-spline bases. In contrast, the present study employs a fourth-order accurate collocation framework constructed with cubic trigonometric B-splines, which provides improved resolution of nonlinear convective effects. The proposed formulation preserves the local support and computational efficiency of spline methods while achieving enhanced accuracy without increasing the stencil width.

First of all, some definitions are presented in Section 2. Then, the discretization of the time-fractional Burgers equation using $L1$ algorithm and Crank-Nicolson approach with the processes obtaining numerical algorithm for the equation is presented in Section 3. In Section 4, the stability of the numerical algorithm which is obtained in Section 3 will be discussed. Lastly, the numerical solutions of the three examples related to time-fractional Burgers equation are investigated and numerical solutions of the problems are presented in Section 5 with tables, error norms and graphs. In Section 6, the conclusion of the study is given.

2 Definitions and preliminaries

In this section, a brief overview of essential definitions and results from fractional calculus is presented.

Definition 1. Riemann-Liouville fractional derivative: Let $f \in L_1(a,t)$ and $\gamma > 0$. The *left-sided Riemann–Liouville fractional derivative* of order γ is defined by

$$ {}^{RL}D_a^{\gamma} f(t) = \frac{1}{\Gamma(n-\gamma)} \frac{d^n}{dt^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{\gamma-n+1}} \, d\tau, $$

where $n = \lceil \gamma \rceil$ and $\Gamma(\cdot)$ denotes the Gamma function [1].

Definition 2. Caputo fractional derivative: Let $f \in C^n(a,t)$ and $\gamma > 0$. The *left-sided Caputo fractional derivative* of order γ is defined by

$$ {}^{C}D_a^{\gamma} f(t) = \frac{1}{\Gamma(n-\gamma)} \int_a^t \frac{f^{(n)}(\tau)}{(t-\tau)^{\gamma-n+1}} \, d\tau, $$

where $n = \lceil \gamma \rceil$, $f^{(n)}$ denotes the n-th derivative of f, and $\Gamma(\cdot)$ denotes the Gamma function [1].

2.1 Trigonometric B-spline functions

A fundamental part of the applied mathematics, approximation problems are the representation of a known or unknown function by means of a well-defined set of special functions that are easy to compute and have certain analytic properties. Algebraic and trigonometric polynomials, exponential functions, multinomial and trigonometric splines are examples of special functions [23]. Trigonometric spline functions were first introduced in 1964 by Schoenberg [24]. Due to their geometric characteristics and shape-preserving properties, their continuous nature, positive definiteness, and partition of unity properties, trigonometric spline functions have attracted considerable attention in image processing, computer graphics, CAM design, trajectory

generation, and many other mathematical and physical fields [25]. The following recursive formula defines trigonometric B-spline functions of degree k. Let $\{x_i\}$ denote a uniform partition of the spatial domain with step size $h = x_{i+1} - x_i$.

$$T_i^k(x) = \frac{\sin(\frac{x-x_i}{2})}{\sin\left(\frac{x_{i+k}-x_i}{2}\right)} T_i^{k-1}(x) + \frac{\sin(\frac{x_{i+k+1}-x}{2})}{\sin\left(\frac{x_{i+k+1}-x_{i+1}}{2}\right)} T_{i+1}^{k-1}(x). \tag{3}$$

Starting with $k = 3$ in the Eq.(3) trigonometric cubic B-spline functions can be obtained as follows

$$T_i^3(x) = \frac{p(x_i)}{\sin(\frac{3h}{2})} T_i^2(x) + \frac{p(x_{i+4})}{\sin(\frac{3h}{2})} T_{i+1}^2(x), \tag{4}$$

where $p(x_i) = \sin(\frac{x-x_i}{2})$. By performing the necessary operations, we have

$$T_i^3(x) = \frac{1}{\theta} \begin{cases} p^3(x_i), & x_i \le x < x_{i+1} \\ -p^2(x_i)p(x_{i+2}) - p(x_i)p(x_{i+3})p(x_{i+1}) - p(x_{i+4})p^2(x_{i+1}), & x_{i+1} \le x < x_{i+2} \\ p(x_i)\ p^2(x_{i+3}) + \ p(x_{i+4})p(x_{i+3})p(x_{i+1}) + p^2(x_{i+4})p(x_{i+2}), & x_{i+2} \le x < x_{i+3} \\ -p^3(x_{i+4}) & x_{i+3} \le x < x_{i+4}, \\ 0 & \text{other cases}, \end{cases}$$

where $\theta = \sin(\frac{h}{2})\sin(h)\sin(\frac{3h}{2})$. Generally, an approximate solution of a given problem using cubic trigonometric cubic B-spline functions can be defined as

$$u(x,t) \approx S(x,t) = \sum_{i=-1}^{N+1} T_i^3(x)\,\delta_i(t). \tag{5}$$

Here, the $u(x,t)$ symbolize the exact solution of the problem and the coefficients δ_i are time-dependent variables, $T_i^3(x)$ functions represent trigonometric cubic B-spline functions. Additionally, the approximate solution and its first and second order derivatives at nodes x_j are

$$\begin{aligned} S(x,t) &= a_1\delta_{j-1} + a_2\delta_j + a_3\delta_{j+1} \\ S'(x,t) &= b_1\delta_{j-1} + b_2\delta_{j+1} \\ S''(x,t) &= c_1\delta_{j-1} + c_2\delta_j + c_3\delta_{j+1}. \end{aligned} \tag{6}$$

Here;

$$\begin{aligned} &a_1 = \sin^2(\tfrac{h}{2})\csc(h)\csc(\tfrac{3h}{2}), a_2 = \tfrac{2}{1+2\cos(h)}, \quad a_1 = a_3 \\ &b_1 = -\tfrac{3\csc(\frac{3h}{2})}{4}, \quad b_2 = \tfrac{3\csc(\frac{3h}{2})}{4} \\ &c_1 = \tfrac{3(3\cos^2(\frac{h}{2})-1)}{4(\sin(h)\sin(\frac{3h}{2}))}, c_2 = -\tfrac{3\cot^2(\frac{h}{2})}{(2+4\cos(h))}, c_3 = c_1. \end{aligned} \tag{7}$$

2.2 Higher-order approach

As defined in the previous subsection, the approximate solution is given. Now, in this subsection, a higher-order nodal approximation of the solution and its derivatives is presented. Suppose that $S(x,t)$ satisfies the following interpolation and boundary conditions for the collocation nodes x_j of the uniform spatial grid, with $j = 0, 1, \cdots, N$.

$$\begin{aligned} &S(x_j,t) = u(x_j,t), && 0 \le j \le N \\ &S''(x_j,t) = u''(x_j,t) - \tfrac{1}{12}h^2u^{(4)}(x_j,t), && j = 0, N. \end{aligned} \tag{8}$$

If $u(x,t)$ is sufficiently smooth function in $[a,b]$ and $S(x,t)$ satisfies the above conditions, then the following relations hold [26, 27]

$$\begin{aligned} &S'(x_j,t) = u'(x_j,t) + O(h^4), && 0 \le j \le N \\ &S''(x_j,t) = u''(x_j,t) - \tfrac{1}{12}h^2u^{(4)}(x_j,t), + O(h^4) && 0 \le j \le N. \end{aligned} \tag{9}$$

Using finite differences and Taylor expansion, the approximate solution and its derivatives at the nodes can be defined as following way

$$\begin{array}{l} S(x_j) = u(x_j), \quad 0 \le j \le N \\ S''(x_j) = u''(x_j), \quad j = 0, N \end{array} \tag{10}$$

and

$$u^{(4)}(x_j) = \frac{S''(x_{j-1}) - 2S''(x_j) + S''(x_{j+1})}{h^2} + O(h^2). \tag{11}$$

Here, one can obtain

$$\begin{array}{ll} j = 1, & u^{(4)}(x_1) = \frac{S''(x_0) - 2S''(x_1) + S''(x_2)}{h^2} + O(h^2), \\ j = 2, & u^{(4)}(x_2) = \frac{S''(x_1) - 2S''(x_2) + S''(x_3)}{h^2} + O(h^2), \end{array} \tag{12}$$

and

$$u^{(4)}(x_0) = 2u^{(4)}(x_1) - u^{(4)}(x_2). \tag{13}$$

When we use the equalities in (12) in the equation given in (13), it can easily be obtained

$$\begin{aligned} u^{(4)}(x_0) &= \frac{2S''(x_0) - 4S''(x_1) + 2S''(x_2) - S''(x_1) + 2S''(x_2) - S''(x_3)}{h^2} \\ &= \frac{2S''(x_0) - 5S''(x_1) + 4S''(x_2) - S''(x_3)}{h^2} + O(h^2). \end{aligned} \tag{14}$$

Similarly, using (11), it yields

$$\begin{array}{ll} j = N-1, & u^{(4)}(x_{N-1}) = \frac{S''(x_{N-2}) - 2S''(x_{N-1}) + S''(x_N)}{h^2}, \\ j = N-2\,, & u^{(4)}(x_{N-2}) = \frac{S''(x_{N-3}) - 2S''(x_{N-2}) + S''(x_{N-1})}{h^2}, \end{array} \tag{15}$$

and using this equalities for

$$u^{(4)}(x_N) = 2u^{(4)}(x_{N-1}) - u^{(4)}(x_{N-2}). \tag{16}$$

We get

$$u^{(4)}(x_N) = \frac{2S''(x_N) - 5S''(x_{N-1}) + 4S''(x_{N-2}) - S''(x_{N-3})}{h^2} + O(h^2). \tag{17}$$

Lastly, we can obtain as following approximations for fourth order derivative;

$$\begin{array}{ll} u^{(4)}(x_j,t) = \frac{2S''(x_0,t) - 5S''(x_1,t) + 4S''(x_2,t) - S''(x_3,t)}{h^2} + O(h^2), & j = 0 \\ u^{(4)}(x_j,t) = \frac{S''(x_{j-1},t) - 2S''(x_j,t) + S''(x_{j+1},t)}{h^2} + O(h^2), & 1 \le j \le N-1 \\ u^{(4)}(x_j,t) = \frac{2S''(x_N,t) - 5S''(x_{N-1},t) + 4S''(x_{N-2},t) - S''(x_{N-3},t)}{h^2} + O(h^2), & j = N. \end{array} \tag{18}$$

Using equations given by (9), we get following approximations for seconds order derivative;

$$\begin{array}{ll} u''(x_0,t) = \frac{14S''(x_0,t) - 5S''(x_1,t) + 4S''(x_2,t) - S''(x_3,t)}{12} + O(h^4), & j = 0, \\ u''(x_j,t) = \frac{S''(x_{j-1},t) + 10S''(x_j,t) + S''(x_{j+1},t)}{12} + O(h^4), & 1 \le j \le N-1 \\ u''(x_N,t) = \frac{14S''(x_N,t) - 5S''(x_{N-1},t) + 4S''(x_{N-2},t) - 5S''(x_{N-3},t)}{12} + O(h^4), & j = N. \end{array} \tag{19}$$

3 Application of the method to time-fractional Burgers equation

This section presents the application of a higher-order trigonometric cubic B-spline collocation method to solve time-fractional Burgers equation with initial and boundary conditions. Let us reconsider the equation with related boundary and initial condition given in (1)-(2). The numerical implementation of the method to the aforementioned equation can be decomposed into 5 key steps. The first one is discretization of the equation

employing the Crank-Nicolson approach for spatial derivatives and $L1$ algorithm for the time fractional order derivative given in [28]. When we apply the first step, we have the following equation

$$\frac{(\Delta t)^{-\gamma}}{\Gamma(2-\gamma)}\sum_{k=0}^{n-1} b_k\left[u^{n-k}-u^{n-k-1}\right]+\frac{(uu_x)^{n+1}+(uu_x)^n}{2}-\nu\frac{(u_{xx}^{n+1})+(u_{xx}^n)}{2}=\frac{f(x_j,t_{n+1})+f(x_j,t_n)}{2},\quad 1\le j\le N-1. \tag{20}$$

The source term $f(x,t)$ is discretized in time using the Crank-Nicolson scheme and evaluated at the collocation points x_j. As the second step, the nonlinear term $(uu_x)^{n+1}$ is linearized using the Rubin-Graves type technique according to the following formula [29,30]

$$(u^p u_x)^{n+1}=(u^p)^n u_x^{n+1}+p(u^{p-1})^n u_x^n u^{n+1}-p(u^p)^n u_x^n. \tag{21}$$

For $p=1$, the Rubin–Graves linearization yields

$$(uu_x)^{n+1}\approx u^n u_x^{n+1}+u_x^n u^{n+1}-u^n u_x^n. \tag{22}$$

Substituting (22) into (20) leads to

$$\begin{aligned} u^{n+1}\left(1+\tfrac{S\psi_1}{2}\right)+u_x^{n+1}\left(\tfrac{S\psi_2}{2}\right)-u_{xx}^{n+1}\left(\tfrac{S\nu}{2}\right)&=u^n+u_{xx}^n\left(\tfrac{S\nu}{2}\right)\\ +\tfrac{S}{2}\left(f(x,t_{n+1})+f(x,t_n)\right)-\textstyle\sum_{k=1}^n b_k\left[u^{n+1-k}-u^{n-k}\right].& \end{aligned} \tag{23}$$

Here, $S=\Delta t^{\gamma}\Gamma(2-\gamma)$, $\psi_1=u_x^n$ and $\psi_2=u^n$. Accordingly, the coefficients multiplying u^{n+1} and u_x^{n+1} are u_x^n and u^n, respectively. Now, the third key step is using approximate solutions in (5) into the discretized equation given in (23). In the light of the previous section, and using approximate solutions, the third step of the numerical implementation is resulted as the following algebraic equation;

$$j=0:\begin{cases} \delta_{-1}^{n+1}\left[a_1(1+\frac{S}{2}\psi_1)+\frac{Sb_1}{2}\psi_2-\frac{S\nu}{24}(14c_1)\right]+\delta_0^{n+1}\left[a_2(1+\frac{S}{2}\psi_1)-\frac{S\nu}{24}(14c_2-5c_1)\right]\\ +\delta_1^{n+1}\left[a_3\left(1+\frac{S}{2}\psi_1\right)+\frac{Sb_2}{2}\psi_2-\frac{S\nu}{24}(14c_3-5c_2+4c_1)\right]+\delta_2^{n+1}\left[-\frac{S\nu}{24}(-5c_3+4c_2-c_1)\right]\\ +\delta_3^{n+1}\left[-\frac{S\nu}{24}(4c_3-c_2)\right]+\delta_4^{n+1}\left[-\frac{S\nu}{24}(-c_3)\right]\\ \quad=\delta_{-1}^{n}\left[a_1+\frac{S\nu}{24}(14c_1)\right]+\delta_0^{n}\left[a_2+\frac{S\nu}{24}(14c_2-5c_1)\right]+\delta_1^{n}\left[a_3+\frac{S\nu}{24}(14c_3-5c_2+4c_1)\right]\\ \quad+\delta_2^{n}\left[\frac{S\nu}{24}(-5c_3+4c_2-c_1)\right]+\delta_3^{n}\left[\frac{S\nu}{24}(4c_3-c_2)\right]+\delta_4^{n}\left[\frac{S\nu}{24}(-c_3)\right]+\frac{S}{2}\left(f(x_0,t_{n+1})+f(x_0,t_n)\right)\\ \quad-\sum_{k=1}^{n} b_k\left[\left(a_1\delta_{-1}^{n+1-k}+a_2\delta_0^{n+1-k}+a_3\delta_1^{n+1-k}\right)-\left(a_1\delta_{-1}^{n-k}+a_2\delta_0^{n-k}+a_3\delta_1^{n-k}\right)\right], \end{cases} \tag{24}$$

and

$$j=1,2,..,N-1:\begin{cases} \delta_{j-2}^{n+1}\left[-\frac{S\nu}{24}c_1\right]+\delta_{j-1}^{n+1}\left[a_1(1+\frac{S}{2}\psi_1)+\frac{Sb_1}{2}\psi_2-\frac{S\nu}{24}(c_2+10c_1)\right]\\ +\delta_j^{n+1}\left[a_2(1+\frac{S}{2}\psi_1)-\frac{S\nu}{24}(c_3+10c_2+c_1)\right]\\ +\delta_{j+1}^{n+1}\left[a_3(1+\frac{S}{2}\psi_1)+\frac{Sb_2}{2}\psi_2-\frac{S\nu}{24}(10c_3+c_2)\right]+\delta_{j+2}^{n+1}\left[-\frac{S\nu}{24}c_3\right]\\ \quad=\delta_{j-2}^{n}\left[\frac{S\nu}{24}c_1\right]+\delta_{j-1}^{n}\left[a_1+\frac{S\nu}{24}(c_2+10c_1)\right]+\delta_j^{n}\left[a_2+\frac{S\nu}{24}(c_3+10c_2+c_1)\right]\\ \quad\delta_{j+1}^{n}\left[a_3+\frac{S\nu}{24}(10c_3+c_2)\right]+\delta_{j+2}^{n}\left[\frac{S\nu}{24}(c_3)\right]+\frac{S}{2}\left(f(x_j,t_{n+1})+f(x_j,t_n)\right)\\ \quad-\sum_{k=1}^{n} b_k\left[\left(a_1\delta_{m-1}^{n+1-k}+a_2\delta_m^{n+1-k}+a_3\delta_{m+1}^{n+1-k}\right)-\left(a_1\delta_{m-1}^{n-k}+a_2\delta_m^{n-k}+a_3\delta_{m+1}^{n-k}\right)\right], \end{cases} \tag{25}$$

and similarly,

$$
j=N:\begin{cases}
\delta_{N-4}^{n+1}\left[\frac{S\nu}{24}c_1\right]+\delta_{N-3}^{n+1}\left[-\frac{S\nu}{24}\left(-c_2+4c_1\right)\right]+\delta_{N-2}^{n+1}\left[-\frac{S\nu}{24}\left(-c_3+4c_2-5c_1\right)\right] \\
+\delta_{N-1}^{n+1}\left[a_1\left(1+\frac{S}{2}\psi_1\right)+\frac{Sb_1}{2}\psi_2-\frac{S\nu}{24}\left(4c_3-5c_2+14c_1\right)\right]+\delta_N^{n+1}\left[a_2(1+\frac{S}{2}\psi_1)-\frac{S\nu}{24}(-5c_3+14c_2)\right] \\
+\delta_{N+1}^{n+1}\left[a_3(1+\frac{S}{2}\psi_1)+\frac{Sb_2}{2}\psi_2-\frac{S\nu}{24}(14c_3)\right] \\
\quad=\delta_{N-4}^{n}\left[-\frac{S\nu}{24}c_1\right]+\delta_{N-3}^{n}\left[\frac{S\nu}{24}\left(-c_2+4c_1\right)\right]\delta_{N-2}^{n}\left[\frac{S\nu}{24}\left(-c_3+4c_2-5c_1\right)\right] \\
\quad+\delta_{N-1}^{n}\left[a_1+\frac{S\nu}{24}\left(4c_3-5c_2+14c_1\right)\right]+\delta_N^{n}\left[a_2+\frac{S\nu}{24}(-5c_3+14c_2)\right] \\
\quad+\delta_{N+1}^{n}\left[a_3+\frac{S\nu}{24}(14c_3)\right]+\ \frac{S}{2}\left(f(x_N,t_{n+1})+f(x_N,t_n)\right) \\
\quad-\sum_{k=1}^{n}b_k\left[\left(a_1\delta_{N-1}^{n+1-k}+a_2\delta_N^{n+1-k}+a_3\delta_{N+1}^{n+1-k}\right)-\left(a_1\delta_{N-1}^{n-k}+a_2\delta_N^{n-k}+a_3\delta_{N+1}^{n-k}\right)\right].
\end{cases}
\tag{26}
$$

The algebraic equations given in (24) – (26) produces an algebraic equation system consisting of $N+1$ equations and $N+3$ unknowns for the values of $j=0,1,\cdots,N$. In order to obtain a solvable system 2 equations must be added, or 2 unknowns must be eliminated from the algebraic equation system. The fourth step of implementation involves the reducing to system into $N+1$ equations and $N+1$ unknown by utilizing the boundary conditions. Eliminating the parameters $\delta_{-1}$and δ_{N+1} yields an algebraic equation system ready to be solved by iteratively. In the sake of seeing clearly, the discretization boundary conditions and removing process can be expressed by the following equations;

$$
\begin{gathered}
u(a,t)=a_1\delta_{-1}+a_2\delta_0+a_3\delta_1=h_1(t) \\
\delta_{-1}=\frac{h_1(t)}{a_1}-\frac{a_2}{a_1}\delta_0-\frac{a_3}{a_1}\delta_1 \\
u\left(b,t\right)=a_1\delta_{N-1}^{n+1-k}+a_2\delta_N^{n+1-k}+a_3\delta_{N+1}^{n+1-k}=h_2\left(t\right) \\
\delta_{N+1}=\frac{h_2(t)}{a_3}-\frac{a_2}{a_3}\delta_N-\frac{a_1}{a_3}\delta_{N-1}.
\end{gathered}
\tag{27}
$$

The equation system obtained above can be written in matrix form as

$$
A\delta^{n+1}=B\delta^n. \tag{28}
$$

As the last step, the initial vector $\left[\delta_0^0,\delta_1^0\delta_2^0,\cdots,\delta_N^0\right]$ will be constructed using the initial condition $u\left(x,0\right)=g\left(x\right)$ of the equation and approximate solution. For $t=0$ in $S(x,t)$, it follows that

$$
\begin{array}{ll}
j=0 & a_1\delta_{-1}+a_2\delta_0+a_3\delta_1=\ g(x_0) \\
j=1 & a_1\delta_0+a_2\delta_1+a_3\delta_2=\ g\left(x_1\right) \\
\vdots & \vdots \\
j=N & a_1\delta_{N-1}+a_2\delta_N+a_3\delta_{N+1}=g(x_N)
\end{array}
\tag{29}
$$

the system in (29) is in the form $N+1$ equations and $N+3$ unknowns. The system given in (29) can be solved after the elimination of the parameters $\delta_{-1}$and δ_{N+1} like the previous step using the Neumann boundary conditions $u_x(a,t)=h_3(t)$ and $u_x\left(b,t\right)=h_4\left(t\right)$. The implementation of the method will be completed via obtaining the initial vector as follows

$$
\begin{array}{ll}
j=0; & a_2\delta_0+\left(a_3-\frac{b_2}{b_1}a_1\right)\delta_1=g(x_0)-\frac{h_3(0)}{b_1}a_1 \\
1\le j\le N-1; & a_1\delta_{j-1}+a_2\delta_j+a_3\delta_{j+1}=g(x_j) \\
j=N; & \left(a_1-\frac{b_1}{b_2}a_3\right)\delta_{N-1}+a_2\delta_N=g(x_N)-\frac{h_4(0)}{b_2}a_3
\end{array}
\tag{30}
$$

Consequently, to obtain numerical solutions, the system given in (24) – (26) is going to be solved iteratively after implementation of the boundary conditions and iteration will be started by the initial vector which is obtained after solving the system given in (30). More clearly, the parameters δ_j^{n+1} are going to be obtained by the parameters δ_j^n at desired time level n. And, using the known values of δ_j^{n+1} in the (5) will produce approximate solutions for time-fractional Burgers equation.

4 Stability analysis

In this section, we are going to perform the stability analysis of the scheme obtained via the application of the suggested method for time-fractional Burgers equation using the von-Neumann criterion and with the help of mathematical induction. According to the von-Neumann criterion, if the amplification factor $|\lambda| \leq 1$, then the error difference between numerical and exact solutions remains bounded as the number of temporal iteration increases. For the stability analysis, the homogenous part of the equation is considered as used in [31]. Consider the numerical scheme which is obtained using the suggested method related with the linearized form of the Eq. (1) by

$$\begin{aligned}
&\delta_{j-2}^{n+1}\left(\tfrac{-S\nu c_1}{24}\right)+\delta_{j-1}^{n+1}\left(a_1+\tfrac{S}{2}\left(\tau b_1-\tfrac{\nu}{12}\left(10c_1+c_2\right)\right)\right)+\delta_j^{n+1}\left(a_2-\tfrac{S\nu}{12}\left(c_1+5c_2\right)\right)\\
&+\delta_{j+1}^{n+1}\left(a_1+\tfrac{S}{2}\left(\tau b_2-\tfrac{\nu}{12}\left(10c_1+c_2\right)\right)\right)+\delta_{j+2}^{n+1}\left(\tfrac{-S\nu c_1}{24}\right)\\
&\quad=\delta_{j-2}^{n}\left(\tfrac{S\nu c_1}{24}\right)+\delta_{j-1}^{n}\left(a_1-\tfrac{S}{2}\left(\tau b_1-\tfrac{\nu}{12}\left(10c_1+c_2\right)\right)\right)+\delta_j^{n}\left(a_2+\tfrac{S\nu}{12}\left(c_1+5c_2\right)\right)\\
&\quad+\delta_{j+1}^{n}\left(a_1-\tfrac{S}{2}\left(\tau b_2-\tfrac{\nu}{12}\left(10c_1+c_2\right)\right)\right)+\delta_{j+2}^{n}\left(\tfrac{S\nu c_1}{24}\right)\\
&\quad-\sum_{k=0}^{n} b_k^{\alpha}\left(\left(a_1\delta_{j-1}^{n+1-k}+a_1\delta_j^{n+1-k}+a_1\delta_{j+1}^{n+1-k}\right)-\left(a_1\delta_{j-1}^{n-k}+a_1\delta_j^{n-k}+a_1\delta_{j+1}^{n-k}\right)\right)
\end{aligned} \tag{31}$$

where $\tau = u_j^n$. Assume that the amplification factor of the Fourier mode is defined as

$$\delta_j^n = \lambda^n e^{ijh\beta} \tag{32}$$

where $i=\sqrt{-1}$ and β is the number of wavelengths. Using (32) in Eq. (31) and dividing $e^{ijh\beta}$ yields following equation

$$\begin{aligned}
&\lambda^{n+1}\left\{\left(\tfrac{-S\nu c_1}{24}\right)\left(e^{-2ih\beta}+e^{2ih\beta}\right)+\left(a_1-\tfrac{S\nu}{24}\left(10c_1+c_2\right)\right)\left(e^{-ih\beta}+e^{ih\beta}\right)\right.\\
&\left.+a_2-\tfrac{S\nu}{12}\left(c_1+5c_2\right)+\tfrac{S}{2}\tau\left(b_1+b_2\right)\right\}\\
&\quad=\lambda^{n}\left\{\left(\tfrac{S\nu c_1}{24}\right)\left(e^{-2ih\beta}+e^{2ih\beta}\right)+\left(a_1+\tfrac{S\nu}{24}\left(10c_1+c_2\right)\right)\left(e^{-ih\beta}+e^{ih\beta}\right)\right.\\
&\quad\left.+a_2+\tfrac{S\nu}{12}\left(c_1+5c_2\right)-\tfrac{S}{2}\tau\left(b_1+b_2\right)\right\}\\
&\quad-\lambda^{n}\sum_{k=0}^{n}b_k^{\alpha}\left(\lambda^{1-k}\left(a_2+a_1\left(e^{-ih\beta}+e^{ih\beta}\right)\right)-\lambda^{-k}\left(a_2+a_1\left(e^{-ih\beta}+e^{ih\beta}\right)\right)\right).
\end{aligned}\tag{33}$$

We know that $e^{-ih\beta}+e^{ih\beta}=2\cos\left(h\beta\right)$, $e^{ih\beta}-e^{-ih\beta}=2i\sin\left(h\beta\right)$, $b_2=-b_1$, grouping the resembling terms, we get

$$\begin{aligned}
&\lambda^{n+1}\left\{2\left(\tfrac{-S\nu c_1}{24}\right)\cos\left(2h\beta\right)+2\left(a_1-\tfrac{S\nu}{24}\left(10c_1+c_2\right)\right)\cos\left(h\beta\right)\right.\\
&\left.+a_2-\tfrac{S\nu}{12}\left(c_1+5c_2\right)-iS\tau b_1\sin\left(h\beta\right)\right\}\\
&\quad=\lambda^{n}\left\{2\left(\tfrac{S\nu c_1}{24}\right)\cos\left(2h\beta\right)+2\left(a_1+\tfrac{S\nu}{24}\left(10c_1+c_2\right)\right)\cos\left(h\beta\right)\right.\\
&\quad\left.+a_2+\tfrac{S\nu}{12}\left(c_1+5c_2\right)+iS\tau b_1\sin\left(h\beta\right)\right\}\\
&\quad-\lambda^{n}\sum_{k=0}^{n}b_k^{\alpha}\left(\lambda^{1-k}\left(2a_1\cos\left(h\beta\right)+a_2\right)-\lambda^{-k}\left(2a_1\cos\left(h\beta\right)+a_2\right)\right).
\end{aligned}\tag{34}$$

After some arrangements, we have

$$\begin{aligned}
&\lambda^{n+1}\left\{2a_1\cos\left(h\beta\right)+a_2-\tfrac{S\nu}{24}\left(2c_1\cos\left(2h\beta\right)+2\left(10c_1+c_2\right)\cos\left(h\beta\right)+\left(2c_1+10c_2\right)\right)-iS\tau b_1\sin\left(h\beta\right)\right\}\\
&\quad=\lambda^{n}\left\{2a_1\cos\left(h\beta\right)+a_2+\tfrac{S\nu}{24}\left(2c_1\cos\left(2h\beta\right)+2\left(10c_1+c_2\right)\cos\left(h\beta\right)+\left(2c_1+10c_2\right)\right)+iS\tau b_1\sin\left(h\beta\right)\right\}\\
&\quad-\lambda^{n}\sum_{k=0}^{n}b_k^{\alpha}\left(\lambda^{1-k}\left(2a_1\cos\left(h\beta\right)+a_2\right)-\lambda^{-k}\left(2a_1\cos\left(h\beta\right)+a_2\right)\right).
\end{aligned}\tag{35}$$

Without the loss of generality, we can assume that $\beta=0$ as in [32, 33], it yields

$$\begin{aligned}
&\lambda^{n+1}\left\{2a_1+a_2-\tfrac{S\nu}{24}\left(2c_1+2\left(10c_1+c_2\right)+\left(2c_1+10c_2\right)\right)\right\}\\
&\quad=\lambda^{n}\left\{2a_1+a_2+\tfrac{S\nu}{24}\left(2c_1+2\left(10c_1+c_2\right)+\left(2c_1+10c_2\right)\right)\right\}
\end{aligned}\tag{36}$$

and

$$\frac{\lambda^{n+1}}{\lambda^{n}}=\frac{2a_1+a_2+\frac{S\nu}{12}\left(2c_1+c_2\right)}{2a_1+a_2-\frac{S\nu}{12}\left(2c_1+c_2\right)}-\frac{\left(2a_1+a_2\right)}{2a_1+a_2-\frac{S\nu}{12}\left(2c_1+c_2\right)}\sum_{k=0}^{n}b_k^{\alpha}\left(\lambda^{1-k}-\lambda^{-k}\right).\tag{37}$$

Now, we apply mathematical induction. For $n=0$, equation (37) becomes

$$\frac{\lambda^{1}}{\lambda^{0}}=\frac{2a_1+a_2+\frac{S\nu}{12}\left(2c_1+c_2\right)}{2a_1+a_2-\frac{S\nu}{12}\left(2c_1+c_2\right)}.$$

For $\left|\frac{\lambda^{n+1}}{\lambda^n}\right| \leq 1$, $\left|2a_1+a_2+\frac{S\nu}{12}(2c_1+c_2)\right| \leq \left|2a_1+a_2-\frac{S\nu}{12}(2c_1+c_2)\right|$ and $2c_1+c_2 \leq 0$. Using Taylor expansion for c_1 and c_2, we get

$$c_1 \approx \frac{1}{h^2}-\frac{1}{10}+\frac{113h^2}{16800}-\frac{197h^4}{739200}+O\left(h^6\right)$$

$$c_2 \approx -\frac{2}{h^2}+\frac{1}{6}-\frac{29h^2}{1080}+\frac{19h^4}{10080}+O\left(h^6\right)$$

and it is clear that for sufficiently small $h(i.e., h \to 0)$, $2c_1+c_2 \leq 0$. Thus, for $n=0$, $\left|\lambda^1\right| \leq \left|\lambda^0\right|$. Now, assume that $\left|\lambda^\ell\right| \leq \left|\lambda^0\right|$ for $\ell = 1,2,\cdots,n$. We now show that it also holds for $\ell = n+1$. Using the assumption $\left|\lambda^\ell\right| \leq \left|\lambda^0\right|$ and $b_k^\alpha = (k+1)^{2-\alpha}-k^{2-\alpha} > 0$ for $0<\alpha<1$, we can conclude from Eq. (37)

$$\left|\frac{\lambda^{n+1}}{\lambda^n}\right| \leq \left|\frac{2a_1+a_2+\frac{S\nu}{12}(2c_1+c_2)}{2a_1+a_2-\frac{S\nu}{12}(2c_1+c_2)}\right|+2\left|\lambda^0\right|\left|\frac{(2a_1+a_2)}{2a_1+a_2-\frac{S\nu}{12}(2c_1+c_2)}\right|\sum_{k=0}^{n} b_k^\alpha. \tag{38}$$

Thus, since $2c_1+c_2 \leq 0$, we get

$$\left|\lambda^{n+1}\right| \leq L\left|\lambda^0\right|$$

where $L = \left|\frac{2a_1+a_2+\frac{S\nu}{12}(2c_1+c_2)}{2a_1+a_2-\frac{S\nu}{12}(2c_1+c_2)}\right|+2\left|\frac{(2a_1+a_2)}{2a_1+a_2-\frac{S\nu}{12}(2c_1+c_2)}\right|\sum_{k=0}^{n} b_k^\alpha$. Hence, the proposed higher-order trigonometric cubic B-spline collocation scheme is unconditionally stable in the sense of the von Neumann criterion.

5 Numerical results

In this section, three numerical examples are presented to validate the accuracy and performance of the proposed higher-order trigonometric cubic B-spline collocation method for the time-fractional Burgers equation. To quantitatively assess the accuracy of the numerical solutions and to compare the obtained results with those reported in the literature, the following error norms are employed:

$$\begin{aligned} L_2 &= \|u(x,t)-S(x,t)\|_2 = \sqrt{h\sum_{j=1}^{N}\left|u(x_j,t_n)-S(x_j,t_n)\right|^2} \\ L_\infty &= \|u(x,t)-S(x,t)\|_\infty = \max_j\left|u(x_j,t_n)-S(x_j,t_n)\right|. \end{aligned} \tag{39}$$

Additionally, the convergence rate of the method is calculated with the formula

$$RoC = \frac{ln(Error(N_2)/Error(N_1))}{ln(N_1/N_2)}, \tag{40}$$

where $Error(N_1)$ and $Error(N_2)$ represent the errors at number of partition N_1 and N_2, respectively. The numerical results are organized into three parts: accuracy tests, comparative studies, and convergence analysis for each problems.

5.1 Example 1

As the first example; consider the time-fractional Burgers equation given by (1) having the related boundary conditions given below. As the first test problem, we consider the time-fractional Burgers equation given by (1) subject to the following initial and boundary conditions:

$$\begin{aligned} &u(0,t) = u(1,t) = 0, && t \in [0,T], \\ &u_x(0,t) = u_x(1,t) = 2\pi t^2, && t \in [0,T], \\ &u(x,0) = 0, && x \in [0,1]. \end{aligned} \tag{41}$$

and the corresponding source term is

$$f(x,t) = \frac{2t^{2-\gamma}\sin(2\pi x)}{\Gamma(3-\gamma)}+2\pi t^4\sin(2\pi x)\cos(2\pi x)+4\nu\pi^2 t^2\sin(2\pi x) \tag{42}$$

and exact solution of the problem is $u(x,t) = t^2 \sin(2\pi x)$ [20]. With the aim of analyzing the first problem, firstly, the time step and viscosity parameter are taken as $\Delta t = 0.0005, \nu = 1$ for varying values of space partition number of N and fractional order derivative. Specifically, the values of fractional derivatives are chosen as $\gamma = 0.25$, 0.5, 0.75, 0.9, respectively and presented in Table 1. Additionally, for a fixed value of $N = 120$, the time step size is varied from $\Delta t = 0.002$ to 0.0005, and the corresponding numerical results are presented in Table 2. Upon examining Table 1, it is evident that increasing the number of spatial partitions has a significant effect on reducing both of the error norms. Notably, for $N = 10$, the infinity norm of the error decreases from $1.55586455 \times 10^{-3}$ to $1.71989088 \times 10^{-5}$, demonstrating a clear reduction trend. Similarly, Table 2 shows that a decrease in time step size generally leads to a reduction in the error norms. Subsequently, comparison tables (Tables 3 and 4) are presented to observe the contribution and effectiveness of the proposed method in solving the problem. In Table 3, for $\gamma = 0.5$ $\Delta t = 0.00025$, the results obtained using the proposed method with different numbers of spatial partitions ($N = 40$,80 and 100) are compared with those from the cubic B-spline finite element method [10] and the trigonometric tension B-spline collocation method [11]. Table 4 presents a comparison of results for $\gamma = 0.5$, $N = 120$ and $t_f = 1$ using different time step sizes $\Delta t = 0.002$, 0.001 and 0.0005, again with references [10] and [11]. Examination of Table 3 reveals that the error norms obtained using the proposed higher-order trigonometric cubic B-spline collocation method are lower than those obtained with the cubic B-spline finite element method and are consistent with the results from [11]. Similarly, Table 4 indicates that for $N = 120$, the error norms obtained using a higher-order trigonometric cubic B-spline collocation method are lower than those obtained using both the cubic B-spline finite element method and the trigonometric tension B-spline collocation method. Moreover, the reduction in error norms becomes more significant as the time step size decreases.

Table 1 The error norms for $\Delta t = 0.0005$ and different values of N and γ of Problem 1.

N	Norm	$\gamma = 0.25$	$\gamma = 0.5$	$\gamma = 0.75$	$\gamma = 0.9$
10	L_2	$1.17608481 \times 10^{-3}$	$1.16808997 \times 10^{-3}$	$1.16122058 \times 10^{-3}$	$1.15977847 \times 10^{-3}$
	L_∞	$1.56691086 \times 10^{-3}$	$1.55586455 \times 10^{-3}$	$1.54633448 \times 10^{-3}$	$1.54428531 \times 10^{-3}$
20	L_2	$3.24966594 \times 10^{-4}$	$3.22510891 \times 10^{-4}$	$3.21299945 \times 10^{-4}$	$3.23144788 \times 10^{-4}$
	L_∞	$4.58600285 \times 10^{-4}$	$4.55139112 \times 10^{-4}$	$4.53437127 \times 10^{-4}$	$4.56047172 \times 10^{-4}$
40	L_2	$7.78030654 \times 10^{-5}$	$7.69389503 \times 10^{-5}$	$7.73445392 \times 10^{-5}$	$8.01209788 \times 10^{-5}$
	L_∞	$1.09833692 \times 10^{-4}$	$1.08615283 \times 10^{-4}$	$1.09189833 \times 10^{-4}$	$1.13110804 \times 10^{-4}$
80	L_2	$1.25916352 \times 10^{-5}$	$1.21469751 \times 10^{-5}$	$1.29784520 \times 10^{-5}$	$1.60002218 \times 10^{-5}$
	L_∞	$1.78288364 \times 10^{-5}$	$1.71989088 \times 10^{-5}$	$1.83758881 \times 10^{-5}$	$2.26544495 \times 10^{-5}$

Table 2 The error norms for $N = 120$ and different values of Δt and γ at $T = 1$ for Problem 1.

Δt	Norm	$\gamma = 0.25$	$\gamma = 0.5$	$\gamma = 0.75$	$\gamma = 0.9$
0.002	L_2	$2.79799013 \times 10^{-5}$	$2.89453357 \times 10^{-5}$	$2.39205481 \times 10^{-5}$	$1.09172130 \times 10^{-5}$
	L_∞	$3.96292257 \times 10^{-5}$	$4.09955552 \times 10^{-5}$	$3.38827254 \times 10^{-5}$	$1.54759469 \times 10^{-5}$
0.001	L_2	$9.09662936 \times 10^{-6}$	$9.70022354 \times 10^{-6}$	$7.53455829 \times 10^{-6}$	$1.19234428 \times 10^{-6}$
	L_∞	$1.28824323 \times 10^{-5}$	$1.37368370 \times 10^{-5}$	$1.06714152 \times 10^{-5}$	$1.69358924 \times 10^{-6}$
0.0005	L_2	$3.57839140 \times 10^{-7}$	$1.18088684 \times 10^{-8}$	$9.03171600 \times 10^{-7}$	$3.97091243 \times 10^{-6}$
	L_∞	$5.05989927 \times 10^{-7}$	$2.09713397 \times 10^{-8}$	$1.27789156 \times 10^{-6}$	$5.62041476 \times 10^{-6}$

Table 3 Comparison of error norms for $\gamma = 0.5$, $\Delta t = 0.00025$, $t_f = 1$ and different values of N for Problem 1.

N	Norm	Proposed method	Ref. [10]	Ref. [11]
40	L_2	8.18×10^{-5}	1.22×10^{-3}	1.60×10^{-5}
	L_∞	1.15×10^{-4}	1.73×10^{-3}	2.63×10^{-5}
80	L_2	1.70×10^{-5}	1.78×10^{-4}	7.72×10^{-6}
	L_∞	2.40×10^{-5}	2.53×10^{-4}	1.34×10^{-5}
100	L_2	9.14×10^{-6}	5.23×10^{-5}	7.24×10^{-6}
	L_∞	1.29×10^{-5}	7.65×10^{-5}	1.19×10^{-5}

Table 4 Comparison of error norms for $\gamma = 0.5$, $N = 120$, $t_f = 1$ and different values of Δt for Problem 1.

Δt	Norm	Proposed method	Ref. [10]	Ref. [11]
0.002	L_2	2.89×10^{-5}	1.22×10^{-3}	5.55×10^{-5}
	L_∞	4.09×10^{-5}	1.72×10^{-3}	8.01×10^{-5}
0.001	L_2	9.70×10^{-6}	5.32×10^{-4}	2.77×10^{-5}
	L_∞	1.37×10^{-5}	7.53×10^{-4}	3.92×10^{-5}
0.0005	L_2	1.18×10^{-8}	1.89×10^{-4}	1.39×10^{-5}
	L_∞	2.09×10^{-8}	2.68×10^{-4}	2.05×10^{-5}

To investigate the convergence rate of the method for Problem 1, the convergence ratios are presented in Table 5 and 6 for $\gamma = 0.5$, $\nu = 1$ and $t_f = 1$, using halved values of both $(b-a)/N$ and Δt, Since both the spatial and temporal discretizations are refined simultaneously, the reported convergence rates represent the overall observed convergence of the fully discrete scheme. In the first table, the number of spatial partitions N is doubled while the Δt values are reduced by a factor of two. Thus, the convergence rates are 4.78, 4.51, 3.31 and 4.27, respectively. In the second table, the number of $N = 10, 20, 40$ is doubled again and similarly the Δt values are decreased by $1/4$, $1/32$ and $1/256$. It is evident from both of the tables that the convergence rate is approximately 4.

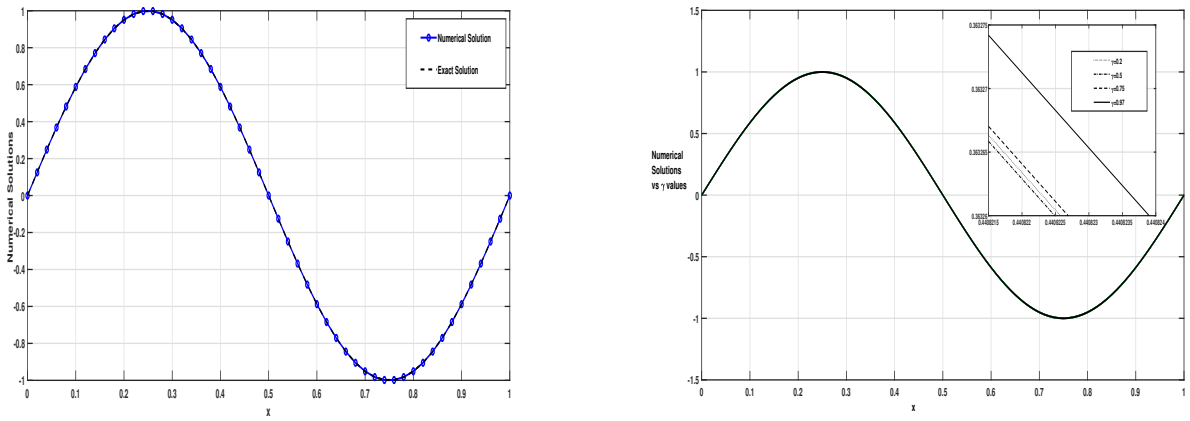
Table 5 Maximum errors and convergence rates for $\gamma = 0.5$, $\nu = 1$, $t_f = 1$ for different values of N and Δt.

N	Δt	L_∞	RoC
6	$1/5$	$4.51974778 \times 10^{-3}$	–
12	$1/20$	$1.64465660 \times 10^{-4}$	4.78
24	$1/80$	$7.17815902 \times 10^{-6}$	4.51
48	$1/320$	$7.21429488 \times 10^{-7}$	3.31
96	$1/1280$	$3.72153127 \times 10^{-8}$	4.27

Table 6 Maximum errors and convergence rates for $\gamma = 0.5$, $\nu = 1$, $t_f = 1$ for different values of N and Δt.

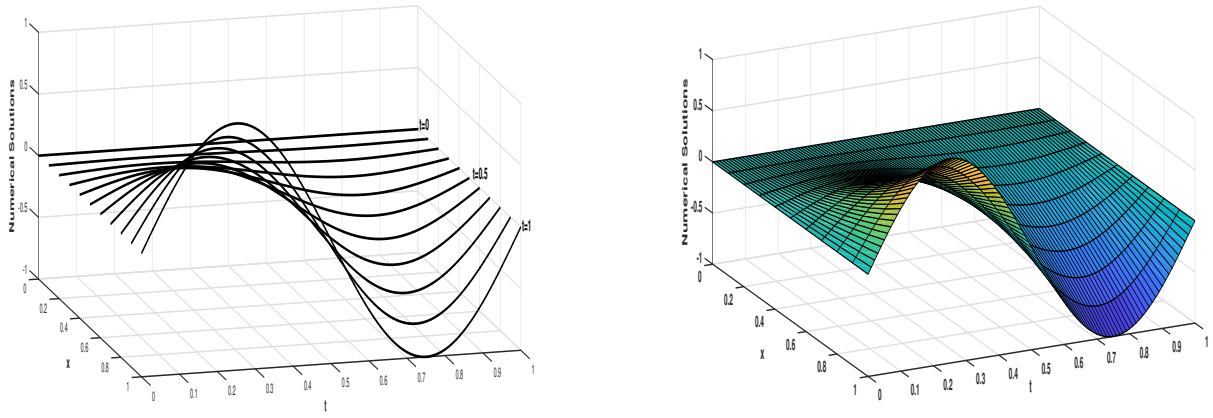
N	Δt	L_∞	RoC
10	$1/4$	$5.26044936 \times 10^{-3}$	–
20	$1/32$	$3.50923812 \times 10^{-4}$	3.90
40	$1/256$	$1.65112329 \times 10^{-5}$	4.40

For example 5.1, we present five figures to illustrate the behavior of the numerical solutions. In Figure 1 (a) involves the graphs of numerical solutions and exact one for $\gamma = 0.5, N = 50$ and $\Delta t = 0.001$ at time $t_f = 1$. It can be seen from the Figure 1 (a), the numerical solutions exhibit the same behavior as the exact solution and two solutions are almost overlapping. In Figure 1 (b), the space partition number and step size are kept as $N = 100$ and $\Delta t = 0.001$ at $t_f = 1$, and the fractional order chosen as $\gamma = 0.2, 0.5, 0.75, 0.97$, respectively. It is clear that a very small change is observed between the solutions with increasing values of γ. For this reason, a zoomed subplot has been inserted in the figure so that the change can be seen. Figure 2 (a) and (b) depict the evolution of the numerical solutions of the time-fractional Burgers equation for $N = 100$, $\gamma = 0.5$, $\Delta t = 0.001$ at time levels $t = 0(0.1)1$. It is seen that with the progression of time, the form of the wave also adapts to its structure. And, lastly in Figure 3, we present the graph of absolute error between the exact and numerical solutions across the spatial domain for $N = 100$, $\gamma = 0.5$, $\Delta t = 0.001$ at $t_f = 1$. It is clear that from Figure 3, the absolute error remains on the order of 10^{-6}, which show high numerical accuracy. One can see a minimum error at $x = 0.5$ and maximum errors are occurred at the wave peaks.



(a) (b)

Fig. 1 Problem 1: Numerical solutions vs exact solutions and γ values.



(a) (b)

Fig. 2 Problem 1: (a) Evolution of numerical solutions vs time levels, (b) 3D graphs of numerical solutions.

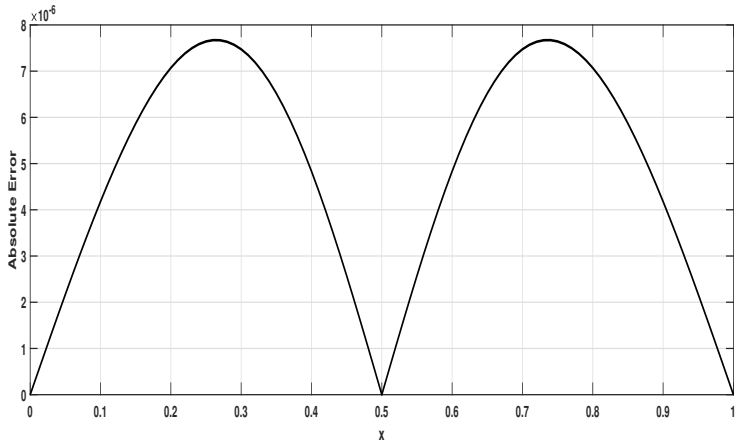


Fig. 3 Problem 1: The absolute errors for $N = 100, \gamma = 0.5, \Delta t = 0.001$ at $t_f = 1$.

5.2 Example 2

As the second example, we consider the time-fractional Burgers equation subject to the following initial and boundary conditions

$$u(0,t) = t^2, u(1,t) = \exp(1)t^2, t \geq 0$$

$$u(x,0) = 0, 0 \leq x \leq 1.$$

The corresponding source term ;

$$f(x,t) = \frac{2t^{2-\gamma}}{\Gamma(3-\gamma)} \exp(x) + t^4 \exp(2x) - \nu t^2 \exp(x)$$

and the exact solution of problem 2 is

$$u(x,t) = t^2 \exp(x).$$

With the similar idea of first example, first of all, to show the effect of fractional order and partition number of space on the error norms, Δt is fixed to 0.0005, viscosity parameter ν, and final time t_f are chosen equal to

1. Then, to show the effect of fractional order and space step size on the error norms L_2 and L_∞, the partition number of space domain is fixed to $N = 120$ for $\nu = 1$, $t_f = 1$. The obtained results of the error norms L_2 and L_∞ are presented in Tables 7 and 8, respectively. Regarding Table 7, the error norms decrease with increasing values of N for each value of fractional order γ. Additionally, it is also observed that for $N = 80$ and $N = 100$, while the error values have an increasing behaviour at first, then a decreasing behavior can be observed on the error norms as the derivative order γ increases. In other words, the derivative order has less negative effect on the error at values close to 0 and 1, whereas the errors increase around $\gamma = 0.5$. In Table 8, as it is expected, for each fixed fractional order γ, the error norms decrease as the number of spatial partitions increases. And, if we consider a fixed value of Δt, a peak on error norms can be seen, however, for different values of fractional order γ, the decrease is again noticeable.

Table 7 The error norms L_2 and L_∞ for varying values of γ and N for $\Delta t = 0.0005$ of Problem 2.

N	Norm	$\gamma = 0.25$	$\gamma = 0.5$	$\gamma = 0.75$	$\gamma = 0.9$
20	L_2	$1.61504086 \times 10^{-4}$	$1.44824109 \times 10^{-4}$	$1.38338571 \times 10^{-4}$	$1.15834339 \times 10^{-4}$
	L_∞	$3.08801388 \times 10^{-4}$	$1.98520497 \times 10^{-4}$	$1.89620989 \times 10^{-4}$	$1.58724458 \times 10^{-4}$
40	L_2	$8.55740423 \times 10^{-5}$	$8.82849416 \times 10^{-5}$	$8.27624785 \times 10^{-5}$	$6.08117897 \times 10^{-5}$
	L_∞	$1.17333257 \times 10^{-4}$	$1.21056287 \times 10^{-4}$	$1.13520309 \times 10^{-4}$	$8.34162090 \times 10^{-5}$
80	L_2	$7.11925979 \times 10^{-5}$	$7.41488030 \times 10^{-5}$	$6.88664654 \times 10^{-5}$	$4.70544770 \times 10^{-5}$
	L_∞	$9.76772696 \times 10^{-5}$	$1.01737613 \times 10^{-4}$	$9.45207070 \times 10^{-5}$	$6.45980625 \times 10^{-5}$
100	L_2	$6.94667643 \times 10^{-5}$	$7.24524065 \times 10^{-5}$	$6.71988959 \times 10^{-5}$	$4.54035631 \times 10^{-5}$
	L_∞	$9.53080411 \times 10^{-5}$	$9.94124999 \times 10^{-5}$	$9.22404235 \times 10^{-5}$	$6.23420557 \times 10^{-5}$

Table 8 The error norms L_2 and L_∞ for varying values of γ and Δt for $N = 120$ of Problem 2.

Δt	Norm	$\gamma = 0.25$	$\gamma = 0.5$	$\gamma = 0.75$	$\gamma = 0.9$
0.002	L_2	$2.66079805 \times 10^{-4}$	$2.75969290 \times 10^{-4}$	$2.46238628 \times 10^{-4}$	$1.54248522 \times 10^{-4}$
	L_∞	$3.65024346 \times 10^{-4}$	$3.78638010 \times 10^{-4}$	$3.37996968 \times 10^{-4}$	$2.11803042 \times 10^{-4}$
0.001	L_2	$1.34644789 \times 10^{-4}$	$1.40213575 \times 10^{-4}$	$1.27686564 \times 10^{-4}$	$8.28012104 \times 10^{-5}$
	L_∞	$1.84721757 \times 10^{-4}$	$1.92384206 \times 10^{-4}$	$1.75274357 \times 10^{-4}$	$1.13702401 \times 10^{-4}$
0.0005	L_2	$6.85292690 \times 10^{-5}$	$7.15309025 \times 10^{-5}$	$6.62930520 \times 10^{-5}$	$4.45067678 \times 10^{-5}$
	L_∞	$9.40182449 \times 10^{-5}$	$9.81468420 \times 10^{-5}$	$9.09997581 \times 10^{-5}$	$6.11154536 \times 10^{-5}$

Table 9 presents a comparison between the proposed method and the previously published method reported in Ref. [10]. for increasing values of the discretization number N and $\gamma = 0.5$, $\Delta t = 0.00025$, $t_f = 1$. From Table 9, it can be seen that as N increases from 40 to 80, both methods exhibit a decreasing behaviour in error norms which is expected. Even so, for both partition numbers $N = 40$ and 80, the proposed method yields lower errors than method used in [10] for the norms L_2 and L_∞. The second comparison table is Table 10 which provides a pointwise comparison of numerical solutions at selected grids, obtained using the collocation method and [12], alongside the exact solution using the error L_∞ for $\gamma = 0.5$, $\Delta t = 0.00025$, $t_f = 1$, $\nu = 1$ and $N = 40$. It can be seen from Table 10, across all data points, proposed collocation method based on trigonometric cubic B-splines yields lower errors than the method used in [12]. We can conclude that the proposed method offers more accurate approximations across the interval.

Table 9 Comparison of errors for $\gamma = 0.5$, $\Delta t = 0.00025$, $t_f = 1$ and different values of N for Problem 2.

N	Method	L_2	L_∞
40	Proposed method	5.36×10^{-5}	7.36×10^{-5}
	Ref. [10]	6.77×10^{-5}	2.09×10^{-4}
80	Proposed method	3.95×10^{-5}	5.42×10^{-5}
	Ref. [10]	4.57×10^{-5}	6.92×10^{-5}

Table 10 Comparison of numerical solutions and errors for $\gamma = 0.5$, $\Delta t = 0.00025$, $t_f = 1$, $\nu = 1$ and $N = 40$ for Problem 2.

x	Proposed method	L_∞	Ref. [12]	L_∞	Exact solution
0.2	1.221366	3.66×10^{-5}	1.221462	5.95×10^{-5}	1.221402
0.4	1.491762	6.24×10^{-5}	1.491934	1.09×10^{-4}	1.491824
0.6	1.822045	7.36×10^{-5}	1.822258	1.39×10^{-4}	1.822118
0.8	2.225480	6.05×10^{-5}	2.225666	1.025×10^{-4}	2.225540

Tables 11 and 12 present the maximum error norms L_∞ and the rates of convergence (RoC) for various combinations of spatial discretization number N and time step size Δt for the values of the parameters $\gamma = 0.9$, $\nu = 1$, and $t_f = 1$. In Table 11, while N increases from 30 to 120 and Δt is diminished from $1/4$ to $1/1024$, the error norms L_∞ decreases from $2.20295625 \times 10^{-2}$ to $1.11304212 \times 10^{-4}$, and the RoC values are approximately 4.19 and 3.46. In Table 12, one can see a different set of partition number as $N = 100, 200$ and 400. For selected values of N and Δt, the error norms L_∞ decreases from $2.20820268 \times 10^{-2}$ to $1.08765049 \times 10^{-4}$ and RoC values remain approximately 4.19 and 3.46. From the tables, it can be concluded that the results obtained demonstrate that the proposed method achieves an observed convergence rate of around 4, and the method is both accurate as well as economical, displaying high-order convergence properties for various spatial and temporal discretizations.

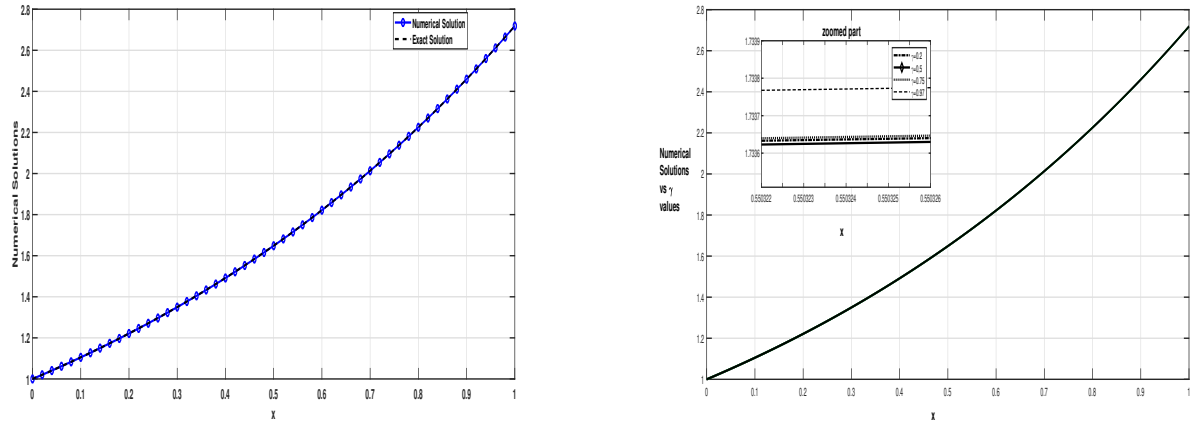
Table 11 The error norms L_∞ and convergence rates for varying values of N and Δt for $\gamma = 0.9$, $\nu = 1$, $t_f = 1$ for Problem 2.

N	Δt	L_∞	RoC
30	$1/4$	$2.20295625 \times 10^{-2}$	–
60	$1/64$	$1.21511503 \times 10^{-3}$	4.18
120	$1/1024$	$1.11304212 \times 10^{-4}$	3.44

Table 12 The error norms L_∞ and convergence rates for varying values of N and Δt for $\gamma = 0.9$, $\nu = 1$, $t_f = 1$ for Problem 2.

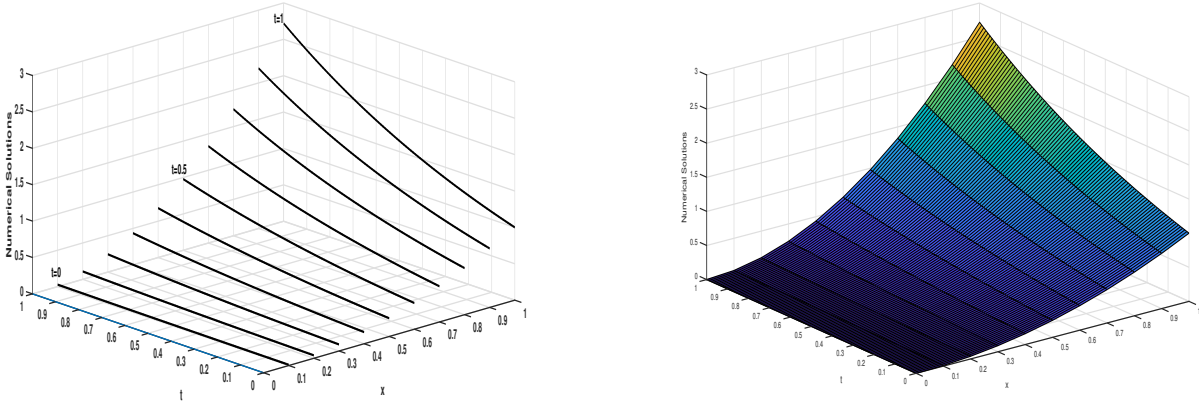
N	Δt	L_∞	RoC
100	$1/4$	$2.20820268 \times 10^{-2}$	–
200	$1/64$	$1.20495216 \times 10^{-3}$	4.19
400	$1/1024$	$1.08765049 \times 10^{-4}$	3.46

For example 5.2, to share the properties of the behaviour of numerical solutions, we present the Figures 4-6. In Figure 4 (a), the values are chosen as $\gamma = 0.75, N = 50$ and $\Delta t = 0.0025$ at time $t_f = 1$ and numerical solutions and exact solutions are plotted on the same axis to show how well they are in agreement with each other. In Figure 4 (b), the partition number of x axis are chosen as $N = 100$, time step size $\Delta t = 0.001$ and the fractional orders are chosen as $\gamma = 0.2$, 0.5, 0.75, 0.97, and numerical solutions of example 5.2 are depicted with a zoomed graph to show the changing values of solutions regarding to fractional order. The behaviour of solutions at different time levels with 3D graphs for $\gamma = 0.5, N = 100$ and $\Delta t = 0.001$ are presented in Figure 5 (a) and (b). Figure 5 (a) is a clear representation of the temporal evolution of the numerical solution profile at different time levels. Figure 5 (b), a 3D surface representation of the numerical solution, provides a helpful view of both spatial and temporal dynamics. And, Figure 6 shows the absolute error distribution between the exact and numerical solutions. It is clear from Figure 6 which shows a maximum absolute error of approximately 1.95×10^{-4} and a peak around $x \approx 0.6$. Since, the error vanishes at the boundaries, it shows that boundary conditions are well respected.



(a) (b)

Fig. 4 Problem 2: (a) Numerical solutions vs exact solutions, (b) Numerical solutions vs γ values.



(a) (b)

Fig. 5 Problem 2: (a) Evolution of numerical solutions vs time levels, (b) 3D graphs of numerical solutions.

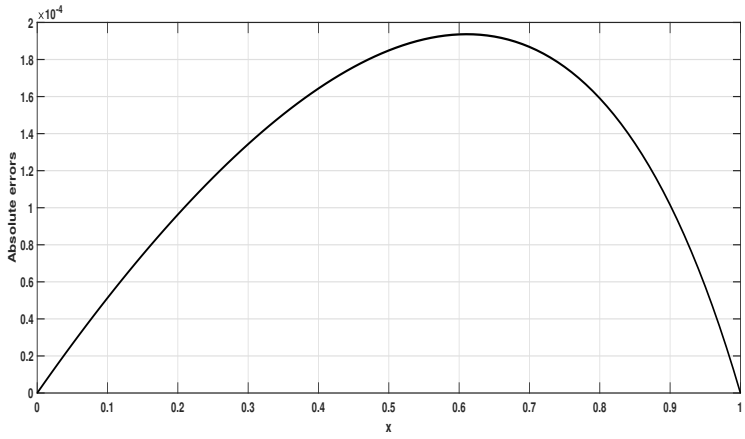


Fig. 6 Problem 2: The absolute errors for $N = 100, \gamma = 0.5, \Delta t = 0.001$ at $t_f = 1$.

5.3 Example 3

As a last example, we are going to consider time-fractional Burgers equation given in (1) via following initial and boundary conditions

$$\begin{aligned} &u(x,0) = 0, \qquad 0 \le x \le 1 \\ &u(0,t) = t^2, u(1,t) = -t^2, \qquad t \ge 0. \end{aligned}$$

For the problem 5.3, the corresponding source term and exact solution are given as

$$\begin{aligned} &f(x,t) = \frac{2t^{2-\gamma}\cos(\pi x)}{\Gamma(3-\gamma)} - \pi t^4 \cos(\pi x)\sin(\pi x) + \nu\pi^2 t^2 \cos(\pi x) \\ &u(x,t) = t^2 \cos(\pi x). \end{aligned}$$

In order to follow the same idea in previous problems, Table 13 and Table 14 are presented to follow the effect of changing fractional order derivative and space partition number and time step size on numerical solutions for fixed values of $\nu = 1, t_f = 1,\ \Delta t = 0.00025$ and $N = 120$, respectively. Table 13 demonstrates a distinct decrease in errors corresponding to an increase in the number of spatial partitions. Furthermore, when analyzing any row with a constant number of divisions, a comparable decline in error is noted as the value of γ rises. Table 14 illustrates an overall trend of diminishing error norms as the time step lowers and the value of γ approaches 1.

Table 13 The error norms L_2 and L_∞ for varying values of γ and N for $\Delta t = 0.00025$ of Problem 3.

N	Norm	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.4$	$\gamma = 0.6$
10	L_2	$3.01229397 \times 10^{-4}$	$3.00321184 \times 10^{-4}$	$2.98502812 \times 10^{-4}$	$2.96792891 \times 10^{-4}$
	L_∞	$4.14249903 \times 10^{-4}$	$4.13024518 \times 10^{-4}$	$4.10575706 \times 10^{-4}$	$4.08282315 \times 10^{-4}$
20	L_2	$7.28223128 \times 10^{-5}$	$7.24982978 \times 10^{-5}$	$7.19129216 \times 10^{-5}$	$7.14959653 \times 10^{-5}$
	L_∞	$1.00695720 \times 10^{-4}$	$1.00239858 \times 10^{-4}$	$9.94134006 \times 10^{-5}$	$9.88181649 \times 10^{-5}$
40	L_2	$1.52632425 \times 10^{-5}$	$1.50860659 \times 10^{-5}$	$1.48106479 \times 10^{-5}$	$1.47187945 \times 10^{-5}$
	L_∞	$2.12934859 \times 10^{-5}$	$2.10460160 \times 10^{-5}$	$2.06611232 \times 10^{-5}$	$2.05322390 \times 10^{-5}$
80	L_2	$8.18623973 \times 10^{-7}$	$6.78279439 \times 10^{-7}$	$4.80627584 \times 10^{-7}$	$4.70380184 \times 10^{-7}$
	L_∞	$1.14201244 \times 10^{-6}$	$9.46133375 \times 10^{-7}$	$6.70126490 \times 10^{-7}$	$6.55416123 \times 10^{-7}$

Table 14 The error norms L_2 and L_∞ for varying values of γ and Δt for $N = 120$ of Problem 3.

Δt	Norm	$\gamma = 0.25$	$\gamma = 0.5$	$\gamma = 0.75$	$\gamma = 0.9$
0.002	L_2	$3.14766794 \times 10^{-5}$	$3.22741925 \times 10^{-5}$	$2.77834353 \times 10^{-5}$	$1.62793905 \times 10^{-5}$
	L_∞	$4.39151250 \times 10^{-5}$	$4.50291619 \times 10^{-5}$	$3.87661038 \times 10^{-5}$	$2.27160596 \times 10^{-5}$
0.001	L_2	$1.46307138 \times 10^{-5}$	$1.51160676 \times 10^{-5}$	$1.31564294 \times 10^{-5}$	$7.53330880 \times 10^{-6}$
	L_∞	$2.04121181 \times 10^{-5}$	$2.10899969 \times 10^{-5}$	$1.83571994 \times 10^{-5}$	$1.05120674 \times 10^{-5}$
0.0005	L_2	$6.23879040 \times 10^{-6}$	$6.51771718 \times 10^{-6}$	$5.66868928 \times 10^{-6}$	$2.93671313 \times 10^{-6}$
	L_∞	$8.70409358 \times 10^{-6}$	$9.09364603 \times 10^{-6}$	$7.90976865 \times 10^{-6}$	$4.09826535 \times 10^{-6}$

Tables 15 and 16 serve as comparative tables. In Table 15, $\gamma = 0.5$, $\Delta t = 0.00025$, and $t_f = 1$ are kept constant, and the spatial partition numbers are taken as 40, 80, and 100. The resultant error norms are compared to those in references [10] and [11]. For $N = 40$, the approach utilized in this study results in error norms comparable to [10] but less than those in [11]. Also, Table 15 clearly shows that as the number of spatial divisions rises, the suggested technique outperformed the other studies. Table 16 uses fixed values of $\gamma = 0.5$, $N = 120$, and $t_f = 1$ for comparing decreasing Δt values. The method produces reduced error norms for each specified Δt compared to those described in [10] and [11].

Table 15 Comparison of errors for $\gamma = 0.5$, $\Delta t = 0.00025$, $t_f = 1$ and different values of N (Problem 3).

N	Method	L_2	L_∞
40	Proposed method	1.47×10^{-5}	2.05×10^{-5}
	Ref. [10]	1.22×10^{-3}	1.73×10^{-3}
	Ref. [11]	1.60×10^{-5}	2.63×10^{-5}
80	Proposed method	4.41×10^{-7}	6.15×10^{-7}
	Ref. [10]	1.78×10^{-4}	2.53×10^{-4}
	Ref. [11]	7.72×10^{-6}	1.34×10^{-5}
100	Proposed method	1.27×10^{-6}	1.77×10^{-6}
	Ref. [10]	5.23×10^{-5}	7.65×10^{-5}
	Ref. [11]	7.24×10^{-6}	1.19×10^{-5}

Table 16 Comparison of errors for $\gamma = 0.5$, $N = 120$, $t_f = 1$ and different values of Δt (Problem 3).

Δt	Method	L_2	L_∞
0.002	Proposed method	3.22×10^{-5}	4.50×10^{-5}
	Ref. [10]	1.22×10^{-3}	1.72×10^{-3}
	Ref. [11]	5.55×10^{-5}	8.01×10^{-5}
0.001	Proposed method	1.51×10^{-5}	2.10×10^{-5}
	Ref. [10]	5.32×10^{-4}	7.53×10^{-4}
	Ref. [11]	2.77×10^{-5}	3.92×10^{-5}
0.0005	Proposed method	6.51×10^{-6}	9.09×10^{-6}
	Ref. [10]	1.89×10^{-4}	2.68×10^{-4}
	Ref. [11]	1.39×10^{-5}	2.05×10^{-5}

Tables 17-19 demonstrate how changing spatial and temporal step sizes influences the convergence order for $\gamma = 0.5$, $v = 1$, and $t_f = 1$. The results demonstrate that the maximum error decreases as the number of spatial divisions increases. Furthermore, achieving a convergence rate close to 4 with finer spatial and temporal discretizations indicates that the use of higher-order basis functions in the construction of the algorithm allows for a better alignment with the solution behavior of the problem.

Table 17 Maximum errors and convergence rates for $\gamma = 0.5$, $v = 1$, $t_f = 1$ and different values of Δt and N (Problem 3).

N	Δt	L_∞	RoC
8	$1/5$	$6.43241077 \times 10^{-3}$	–
16	$1/40$	$4.49763147 \times 10^{-4}$	3.83
32	$1/320$	$3.33582433 \times 10^{-5}$	3.75

Table 18 Maximum errors and convergence rates for $\gamma = 0.5$, $v = 1$, $t_f = 1$ and different values of Δt and N (Problem 3).

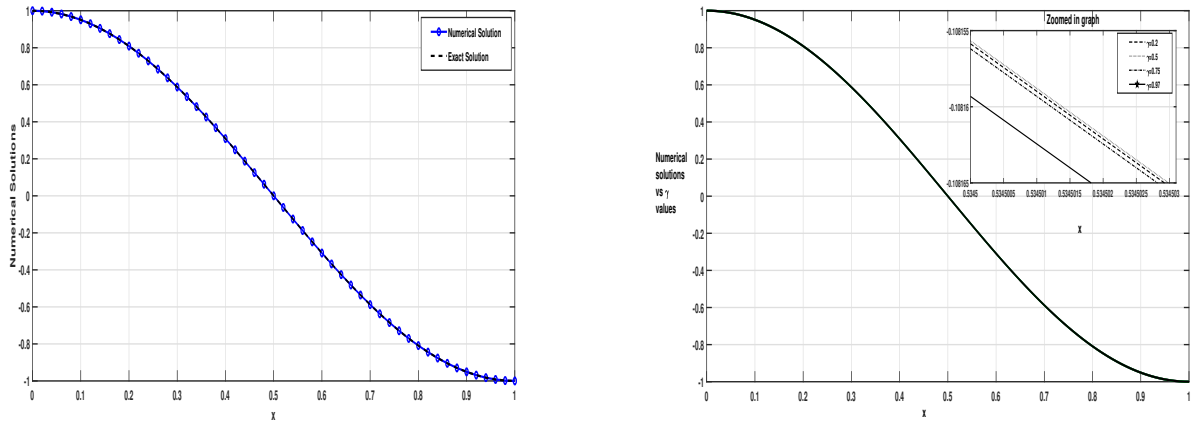
N	Δt	L_∞	RoC
5	$1/4$	$7.90193068 \times 10^{-3}$	–
10	$1/32$	$3.54754064 \times 10^{-4}$	4.47
20	$1/256$	$1.22189859 \times 10^{-5}$	4.85

Table 19 Maximum errors and convergence rates for $\gamma = 0.5$, $v = 1$, $t_f = 1$ and different values of Δt and N (Problem 3).

N	Δt	L_∞	RoC
12	$1/5$	$1.64465660 \times 10^{-4}$	–
24	$1/40$	$7.17815902 \times 10^{-6}$	3.64
48	$1/320$	$7.21429488 \times 10^{-7}$	3.25
96	$1/2560$	$3.72153127 \times 10^{-8}$	3.55

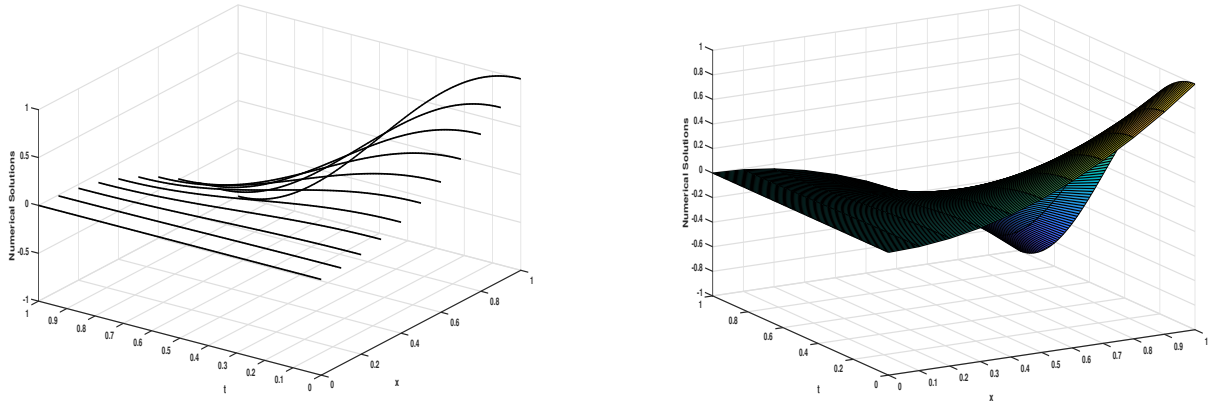
For the problem 5.3, in order to reiterate the visual aims of problem 1 and problem 2, Figure 7 (a) is given

to show the correspondence between the numerical solution and the exact solution for problem 3 for the values $\gamma = 0.5$, $\nu = 1$, $N = 100$ and $\Delta t = 0.001$. Then, in Figure 7 (b), the change of the fractional derivative on the curve is shown with a zoomed image by choosing $N = 100, \Delta t = 0.001, t_f = 1$ for the change of the fractional order $\gamma = 0.2$, 0.5, 0.75, and, 0.97. The temporal and temporal-spatial evolutions of the nature of the solution are shown in Figure 8 (a) and (b) for $\gamma = 0.5$, $N = 100$ and $\Delta t = 0.001$ in 3D plots. At the end, absolute error distribution between the exact and numerical solutions are presented in Figure 9. The absolute error shows a similar behaviour to the absolute error in the first problem, and the vanishing of the error at the boundaries again indicates that the boundary conditions are well provided. From a computational perspective, the proposed method retains the local support property of cubic trigonometric B-splines, leading to sparse and banded system matrices. Therefore, the computational cost per time step can be of the same order as that of existing spline collocation methods. Moreover, the higher-order spatial accuracy enables higher accuracy on relatively coarse grids, resulting in improved efficiency in terms of accuracy versus computational effort.



(a) (b)

Fig. 7 Problem 3: (a) Numerical solutions vs exact solutions, (b) Numerical solutions vs γ values.



(a) (b)

Fig. 8 Problem 3: (a) Evolution of numerical solutions vs time levels, (b) 3D graphs of numerical solutions.

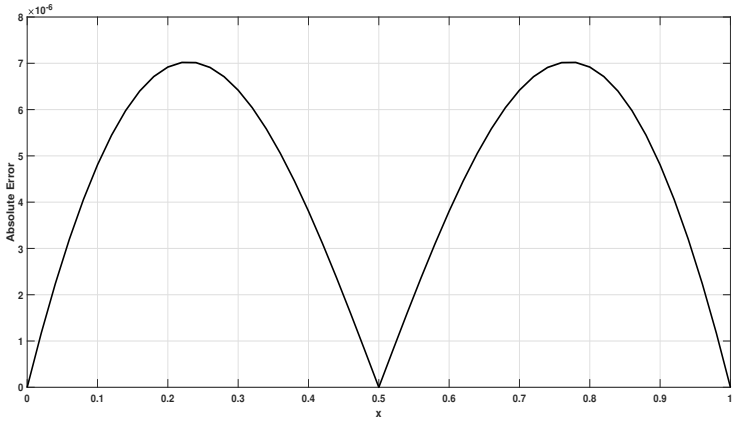


Fig. 9 Problem 3: The absolute Errors for $N = 100, \gamma = 0.5, \Delta t = 0.001$ at $t_f = 1$.

6 Conclusions

In the current article, we have designed and implemented a numerical approach for solving the time-fractional Burgers equation using a higher-order trigonometric cubic B-spline collocation method. The numerical experiments showed that the suggested technique has a convergence rate around four, suggesting a high degree of accuracy. The method's efficacy was further tested by doing error analysis and comparing the results to those of existing methodologies. The smoothness and local support of the trigonometric cubic B-spline basis functions contributed to the scheme's stability and accuracy, making it well-suited for dealing with fractional-order differential equations, which are nonlinear and nonlocal. These properties emphasize the potential of the method for larger applications in complicated physical models driven by fractional dynamics. In future, we hope to investigate the applicability of our technique to various types of fractional partial differential equations. Furthermore, we have intended to study how other types of basis functions such as hyperbolic, polynomial, or wavelet-based splines affect the method's accuracy, computing efficiency, and resilience. These investigations will help to further the development of efficient numerical tools for fractional calculus and its applications.

7 Declarations

7.1 Conflict of interest:

Not applicable.

7.2 Funding:

This work was supported by Research Fund of İnönü University via project number: FBA-2026-4241.

7.3 Author's contribution:

M.Ö.-Resources, Validation, Conceptualization, Formal Analysis, Data curation, Investigation, Visualization. B.K.-Writing-Original Draft, Methodology, Validation, Conceptualization, Formal Analysis, Data curation, Investigation, Visualization. A.E.-Methodology, Validation, Conceptualization, Formal Analysis, Data curation, Investigation, Visualization. All authors read and approved the final submitted version of this manuscript.

7.4 Acknowledgement:

Not applicable.

7.5 Data availability statement:

All data that support the findings of this study are included within the article.

7.6 Usage of AI tools:

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

References

[1] Podlubny I., Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications, Academic Press, USA, 1999.
[2] Ricardo H.J., A Modern Introduction to Differential Equations, Academic Press, USA, 2021.
[3] Qureshi S., Yusuf A., Mathematical modeling for the impacts of deforestation on wildlife species using Caputo differential operator, Chaos Solitons and Fractals, 126, 32–40, 2019.
[4] Miller K.S., Ross B., An Introduction to the Fractional Calculus and Fractional Differential Equations, Wiley-Interscience, USA, 1993.
[5] Caputo M., Linear models of dissipation whose Q is almost frequency independent, Annals of Geophysics, 19(4), 383–393, 1966.
[6] Jain S., Abuomar M., Numerical investigation of the fractional Riccati equation: Analysis, discretization, and applications, Communications in Nonlinear Science and Numerical Simulation, 154(109539), 2026.
[7] Owolabi K.M., Jain S., From spiral turbulence to spatiotemporal chaos: Numerical and analytical study of a novel CFGL system, Chaos Solitons and Fractals, 200(2), 117073, 2025.
[8] Jain S., Owolabi K.M., Pindza E., Mare E., Dynamic complexity in fractional multispecies ecological systems: A Caputo derivative approach, Partial Differential Equations in Applied Mathematics, 16(101293), 2025.
[9] Jain S., Rababah A., Dynamical analysis of fractional-order Burger–Huxley equation using efficient numerical methods, The European Physical Journal Special Topics, 232(14), 2567–2574, 2023.
[10] Esen A., Tasbozan O., Numerical solution of time fractional Burgers equation by cubic B-spline finite elements, Mediterranean Journal of Mathematics, 13, 1325–1337, 2016.
[11] Singh B.K., Gupta M., Trigonometric tension B-spline collocation approximations for time fractional Burgers equation, Journal of Ocean Engineering and Science, 9(5), 508–516, 2024.
[12] Duangpan A., Boonklurb R., Treeyaprasert T., Finite integration method with shifted Chebyshev polynomials for solving time-fractional Burgers equation, Mathematics, 7(12), 1201, 2019.
[13] Tamsir M., et al., A hybrid B-spline collocation technique for the Caputo time fractional nonlinear Burgers equation, Beni-Suef University Journal of Basic and Applied Sciences, 12(1), 95, 2023.
[14] Jeon Y., Bu S., Numerical approach for time-fractional Burgers equation via a combination of Adams–Moulton and linearized technique, Journal of Mathematical Chemistry, 62(5), 1189–1208, 2024.
[15] Hashmi M.S., et al., Cubic spline based differential quadrature method: a numerical approach for fractional Burger equation, Results in Physics, 26, 104415, 2021.
[16] Sadri K., et al., Bivariate Jacobi polynomials depending on four parameters and their effect on solutions of time-fractional Burgers equation, Journal of Computational Science, 83, 102450, 2024.
[17] Yaseen M., Abbas M., An efficient computational technique based on cubic trigonometric B-splines for time fractional Burgers equation, International Journal of Computer Mathematics, 97(3), 725–738, 2020.
[18] Al-Saedi A.A., Rashidinia J., Application of the B-spline Galerkin approach for approximating the time-fractional Burgers equation, Electronic Research Archive, 31(7), 4248–4265, 2023.
[19] Maji S., Natesan S., Error analysis for discontinuous Galerkin method for time-fractional Burgers equation, Mathematical Methods in the Applied Sciences, 47(12), 9703–9717, 2024.
[20] Roul P., Rohil V., A high-accuracy computational technique based on $L2$–1σ and B-spline schemes for solving the nonlinear time-fractional Burgers equation, Soft Computing, 28(7), 6153–6169, 2024.
[21] Taghipour M., Aminikhah H., Application of Pell collocation method for solving the general form of time-fractional Burgers equation, Mathematical Sciences, 17(2), 183–201, 2023.
[22] El-Danaf T.S., Hadhoud A.R., Parametric spline functions for the solution of the one time fractional Burgers equation, Applied Mathematical Modelling, 36(10), 4557–4564, 2012.
[23] Denysiuk V., Polynomial and trigonometric splines, arXiv:2110.04781, 2021.

[24] Schoenberg I.J., On trigonometric spline interpolation, Journal of Mathematics and Mechanics, 13, 795–825, 1964.
[25] Noureen R., Asgir M., Iqbal M.K., Azeem M., Approximate solution of time-fractional gas dynamics equation using exponential B-spline functions, Optical and Quantum Electronics, 57(3), 197, 2025.
[26] Roul P., Goura V.M.K.P., A high-order B-spline collocation scheme for solving a nonhomogeneous time-fractional diffusion equation, Mathematical Methods in the Applied Sciences, 44(1), 546–567, 2021.
[27] Lucas T.R., Error bounds for interpolating cubic splines under various end conditions, SIAM Journal on Numerical Analysis, 11(3), 569–584, 1974.
[28] Oldham K.B., Spanier J., The Fractional Calculus, Academic Press, USA, 1974.
[29] Rubin S.G., Graves R.A., A Cubic Spline Approximation for Problems in Fluid Mechanics, NASA, USA, 1975.
[30] Rashidinia J., Rasoulizadeh M.N., Numerical methods based on radial basis function-generated finite difference (RBF-FD) for solution of GKdVB equation, Wave Motion, 90, 152–167, 2019.
[31] Shafiq M., Abbas M., Abualnaja K.M., Huntul M.J., Majeed A., Nazir T., An efficient technique based on cubic B-spline functions for solving time-fractional advection diffusion equation involving Atangana–Baleanu derivative, Engineering Computations, 38, 901–917, 2022.
[32] 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, 44(1), 901–916, 2021.
[33] Sweilam N.H., Al-Mekhlafi S.M., Albalawi A.O., A novel variable-order fractional nonlinear Klein–Gordon model: a numerical approach, Numerical Methods for Partial Differential Equations, 35(5), 1617–1629, 2019.