Journal of Nonlinear Modeling and Analysis
Volume 7, Number 1, February 2025, 318–333
https://journal.global-sci.org/jnma
DOI:10.12150/jnma.2025.318

Numerical Solutions for Fractional Burgers' Equation Based on Laplace Transform*

Weiye Sun[1], Yulian An[1], Yijin Gao[1,†] and Songting Luo[2]

Abstract The Burgers' equation has widespread applications across various fields. In this paper, we propose an efficient approach for obtaining the numerical solution to the time-fractional Burgers' equation. We extend the classical Burgers' equation to its fractional form by introducing Caputo derivatives. Using the Cole-Hopf transform, we reformulate the problem into a fractional diffusion equation. The Laplace transform method is then applied to convert the equation into an ordinary differential equation (ODE), which can be solved analytically. However, due to the lack of an inverse Laplace transform for this specific form, numerical approximation methods are then utilised to approximate the true solution. Numerical simulations are provided to demonstrate the stability and accuracy of the proposed method.

Keywords Fractional Burgers' equation, Laplace transform, Caputo derivative, numerical simulations

MSC(2010) 35R11, 65D15.

1. Introduction

Burgers' equation is a fundamental mathematical model extensively used in various fields, including fluid dynamics, traffic flow, and non-linear wave propagation in physics, chemistry, and engineering. Its importance lies in its ability to capture the interaction between non-linear convection and diffusion processes, making it crucial for understanding complex physical phenomena. However, despite its significance, a general analytical solution for this complex system remains elusive, prompting researchers to investigate various numerical algorithms for effective solutions.

Numerous numerical methods have been employed to solve Burgers' equation, including the Finite-Difference Method (FDM), Method of Lines (MOL), Finite-Element Method (FEM), and spline techniques, as highlighted by Bonkile *et al.* [4]. Among FDMs, a key approach involves transforming Burgers' equation into the heat equation using the Hopf-Cole transformation. For example, Kutluay *et al.* converted Burgers' equation into a heat diffusion equation and applied explicit and exact-explicit finite-difference methods to solve the transformed equations under

†the corresponding author.
Email address:weiyesunny@163.com(W. Sun), anyl@shisu.edu.cn(Y. An), yjgaomath@163.com(Y. Gao), luos@iastate.edu(S. Luo)
[1]School of Economics and Finance, Shanghai International Studies University, Shanghai, 201620, PR China
[2]Department of Mathematics, Iowa State University, Ames, IA, 50011, USA
*The authors were supported by National Natural Science Foundation of China (grant numbers: 12071302), and Mentor Academic Guidance Program of Shanghai International Studies University (grant number: 2022113028).

specific boundary conditions [23]. Other studies have also leveraged the Hopf-Cole transformation to derive a linear heat equation, yielding promising results [19, 50]. In two-dimensional cases, Bahadir *et al.* introduced a fully implicit finite-difference scheme, solving the non-linear system using Newton's method [3]. Srivastava *et al.* developed a finite-difference technique for coupled viscous Burgers' equations on a uniform grid [48]. The method of lines, initially proposed by Rothe [39], has proven effective in transforming partial differential equations into ordinary differential equation initial value problems. In the context of FEM and spline approaches, Roul *et al.* employed sextic B-spline basis functions for spatial discretisation, achieving highly accurate results with reduced computational time [40], while Majeed *et al.* utilised an extended cubic B-spline collocation scheme for the time-fractional modified Burgers' equation with Caputo fractional derivatives [32]. Dhawan *et al.* provided a comprehensive review of techniques addressing the challenges posed by the non-linear nature of Burgers' equation [8]. Additionally, Cengizci *et al.* stabilized finite element formulations with shock-capturing techniques [6], and Singh *et al.* [46] and Jiwari *et al.* [18] investigated efficient and hybrid methodologies, respectively.

Fractional-order calculus, which extends differentiation to non-integer orders, has gained significant traction in areas such as signal processing, control systems, and mathematical modelling. This is largely due to the time-memory characteristics inherent in fractional derivatives, enabling more accurate modelling of dynamic systems with memory effects. Notable formulations, including the Grünwald-Letnikov [26, 29], Riemann-Liouville [35], and Caputo [10, 31] formulas, are commonly applied. These fractional derivatives have been successfully integrated into ordinary and partial differential equations (ODEs and PDEs), offering new perspectives for addressing complex problems [38, 47].

Incorporating time-fractional derivatives into Burgers' equation allows for the inclusion of memory effects and non-local interactions, which are often present in real-world scenarios but are overlooked in classical models. Analytical solutions to the fractional Burgers' equation are typically limited to specific values of the fractional order parameter, denoted as α. Consequently, efficient numerical methods have been developed to address this limitation, with various studies demonstrating their effectiveness [22, 51]. Finite difference methods have shown particular promise in tackling both time-fractional and space-fractional PDEs [13, 25]. For instance, Chen *et al.* introduced a Fourier method for solving fractional diffusion equations, demonstrating the stability and convergence of their implicit difference approximation scheme [7]. For testing convergence and stability, Roul *et al.* [41] demonstrated the unconditional stability of a proposed non-standard finite difference scheme for the fractional neutron point kinetic equation and utilised Von-Neumann stability analysis for a numerical method applied to the fractional neutron diffusion equation [42]. Other approaches, including finite element methods [17, 27], wavelet methods [49, 52], variational iteration methods [16, 36], homotopy perturbation methods [34], matrix approaches [15, 28], and emerging machine learning techniques [12, 30], have further advanced the solutions for fractional PDEs.

This paper presents a novel approach to numerically solving the fractional Burgers' equation using the Laplace transform. The Laplace transform enables the derivation of an exact solution by directly applying the transform and analytically solving the corresponding ODE. In cases where the inverse Laplace transform lacks an exact solution, we resort to numerical algorithms for computation. The structure of this paper is as follows: Section 2 introduces fractional derivatives, Burgers' equation, and other key definitions. Section 3 applies the Laplace transform to the fractional PDEs, transforming the problem into solvable ODEs. In Section 4, we solve the ODEs and reconstruct the original solution using numerical methods. Section 5 presents numerical examples to validate the theoretical results. Section 6 presents convergence and stability analysis of the proposed model, and Section 7 concludes the paper.

2. Definitions and basic model

2.1. Fractional derivatives

The Caputo fractional calculus has proven highly effective in modelling systems with non-zero initial conditions, making it a suitable choice for this problem. Therefore, we introduce the Caputo fractional derivative into Burgers' equation. The Caputo fractional derivative, as defined in [33], is expressed as:

$$ {}_{t_0}^{C}\mathscr{D}_t^{\alpha} y(t) = \frac{1}{\Gamma(m-a)} \int_{t_0}^{t} \frac{y^{(m)}(\tau)}{(t-\tau)^{1+\alpha-m}} d\tau, \tag{2.1} $$

where $\Gamma(\cdot)$ refers to the Gamma function. For a positive integer n, $\Gamma(n) = (n-1)!$; and for non-integer values, $\Gamma(z)$ is defined as:

$$ \Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} dt. \tag{2.2} $$

The Caputo Fractional Integral is defined as:

$$ {}_{t_0}^{C}\mathscr{D}_t^{-\gamma} y(t) = \frac{1}{\Gamma(\gamma)} \int_{t_0}^{t} \frac{y(\tau)}{(t-\tau)^{1-\gamma}} d\tau. \tag{2.3} $$

2.2. Fractional Burgers' equation

In this paper, our objective is to address the fractional Burgers' equation, as presented in [22]:

$$ \begin{cases} \dfrac{\partial^{\alpha} u}{\partial t^{\alpha}} + u\dfrac{\partial u}{\partial x} - \upsilon\dfrac{\partial^2 u}{\partial x^2} = 0, \quad 0 < x < 1, \quad t > 0; \\ u(x,0) = \sin(\pi x), \quad u(0,t) = u(1,t) = 0, \end{cases} \tag{2.4} $$

where $\frac{\partial^{\alpha} u}{\partial t^{\alpha}}$ is the Caputo fractional derivative mentioned in equation (2.1), and the fractional order is denoted by $\alpha \in (0, 1]$. Here, υ is a constant that implies the given limit of diffusivity and is set as $\upsilon = 1$.

2.3. Cole-Hopf transform

Instead of seeking a direct solution, we apply the Cole-Hopf Transform (CHT) to simplify the equation. The CHT facilitates the reduction of the problem to the fractional heat equation, offering a more tractable form. This transformation has been widely employed in solving complex differential equations encountered in physics and chemistry [11,43,44]. By using the CHT, the Burgers' equation can be rewritten as:

$$u_t + uu_x = \upsilon u_{xx}, \quad t \geq 0. \tag{2.5}$$

To solve this equation, we apply a non-linear change of variables. Following the CHT, we introduce a potential function $\phi(x,t)$ such that $u = \phi_x$. Consequently, the original equation transforms into:

$$\phi_{xt} + \phi_x \phi_{xx} = \upsilon \phi_{xxx},$$

and it can be integrated by x to be:

$$\phi_t + \frac{\phi_x^2}{2} = \upsilon \phi_{xx}.$$

Suppose the new variable ϕ is given by :

$$\phi = -2\upsilon \log(\psi), \tag{2.6}$$

which meets:

$$\begin{cases} \phi_t = -\dfrac{2\upsilon}{\psi}\psi_t, \\ \phi_x = -\dfrac{2\upsilon}{\psi}\psi_x, \\ \phi_{xx} = -\dfrac{2\upsilon}{\psi}\psi_{xx} + 2\upsilon(\dfrac{\psi_x}{\psi})^2, \end{cases}$$

then the transformed equation is:

$$\psi_t = \upsilon \psi_{xx}, \tag{2.7}$$

and equation (2.4) can be modified into the form of fractional heat equation:

$$\begin{cases} \dfrac{\partial^\alpha q}{\partial t^\alpha} = \upsilon \dfrac{\partial^2 q}{\partial x^2}, \\ q(x,0) = e^{-\frac{1-\cos(\pi x)}{2\upsilon\pi}}, \\ q_x(0,t) = q_x(1,t) = 0. \end{cases} \tag{2.8}$$

3. Fractional order partial differential equations (FOPDE) and Laplace transform

3.1. Clarification of interchange of variables

The Laplace transform of a function $f(t)$ is defined as:

$$\mathcal{L}\{f(t)\} = F(s) = \int_0^\infty e^{-st} f(t) dt. \tag{3.1}$$

The Laplace transform converts a function from the time domain (t) to the Laplace domain (s). During this transformation, the variable x remains constant, while t, which is transformed into s, becomes the primary variable. However, when solving the resulting ordinary differential equations, x is treated as the primary variable, and s is determined by t at the initial stage of the calculation. This distinction underscores the dual roles played by x and t during the two phases of the analysis. The transformation of the main integration variable is essential when solving PDEs using the Laplace transform. Thus, in this context, x, t, and s represent variables, while constants are denoted as $\overline{x}$ and $\overline{s}$.

3.2. Laplace-transformed Caputo FOPDE and initial conditions

Several transform methods for solving PDEs exist in the literature. In this paper, we focus on the Laplace transform method, which reshapes PDEs into corresponding ODEs. The Laplace-transformed Caputo derivative is given by [5]:

$$\mathscr{L}\left\{\frac{\partial^\alpha q}{\partial t^\alpha}\right\} = \mathscr{L}\left\{{}_{t_0}^{C}\mathscr{D}_t^\alpha q(\overline{x},t)\right\} = s^\alpha Q(\overline{x},s) - \sum_{k=0}^{n-1} s^{\alpha-k-1} q^{(k)}(0), \tag{3.2}$$

where $Q(s)$ is the Laplace-transformed $q(x,t)$, and according to the properties of the Laplace transform, we get the transformed $q_{xx}(x,t)$ from $Q_{xx}(x,\overline{s})$:

$$\mathscr{L}(q_{xx}(x,t)) = \mathscr{L}\left\{\frac{\partial^2 q(x,t)}{\partial x^2}\right\} = \frac{\partial^2 Q(x,\overline{s})}{\partial x^2} = Q_{xx}(x,\overline{s}). \tag{3.3}$$

Then, the heat equation (2.8) could be Laplace transformed as:

$$\overline{s}^\alpha Q(x,\overline{s}) - \sum_{k=0}^{n-1} \overline{s}^{\alpha-k-1} q^{(k)}(0) = \upsilon \frac{\partial^2 Q(x,\overline{s})}{dx^2}, \tag{3.4}$$

which is arranged to be a constant coefficient non-homogeneous fractional order ODE:

$$\frac{d^2 Q(x,\overline{s})}{dx^2} - \frac{\overline{s}^\alpha}{\upsilon} Q(x,\overline{s}) = \frac{-\overline{s}^{\alpha-1}}{\upsilon} e^{-\frac{1-\cos(\pi x)}{2\upsilon\pi}}. \tag{3.5}$$

Once $\alpha = 1$, we have Laplace transformed Burgers' equation as:

$$\upsilon \frac{d^2 Q(x,\overline{s})}{\partial x^2} - \overline{s} Q(x,\overline{s}) = -e^{-\frac{1-\cos(\pi x)}{2\upsilon\pi}}, \tag{3.6}$$

that could be an examiner for the feasibility of the model.

In the process of solving PDEs, the Laplace transform is frequently employed, particularly in cases involving second-order derivatives, as it is well-suited for obtaining analytical solutions. However, when non-integer orders are introduced, the right-hand side of the equation becomes intertwined with the variable s, which introduces considerable complexity, posing significant challenges to finding analytical solutions.

4. Solving the transformed equations

4.1. Solving the Laplace-transformed ODE

Since the Laplace-transformed equation (3.5) is an ODE with respect to x for each value of s, s can be treated as a constant within the ODE. Accordingly, we solve the ODE by deriving its general solution and one of its particular solutions. The general solution for ODEs of this form is:

$$Q_g = c_1 e^{\sqrt{\frac{s^\alpha}{\upsilon}}x} + c_2 e^{-\sqrt{\frac{s^\alpha}{\upsilon}}x}, \tag{4.1}$$

and one of its particular solutions is:

$$Q_p = e^{-\sqrt{\frac{s^\alpha}{\upsilon}}x}\int_0^x -\frac{e^{\xi\sqrt{\frac{s^\alpha}{\upsilon}}}f(\xi)}{2\sqrt{\frac{s^\alpha}{\upsilon}}}d\xi + e^{\sqrt{\frac{s^\alpha}{\upsilon}}x}\int_0^x \frac{e^{-\xi\sqrt{\frac{s^\alpha}{\upsilon}}}f(\xi)}{2\sqrt{\frac{s^\alpha}{\upsilon}}}d\xi, \tag{4.2}$$

where

$$f(x) = -\frac{s^{\alpha-1}}{\upsilon}e^{-\frac{1-\cos\pi x}{2\upsilon\pi}}.$$

The solution to the transformed equation is the sum of the general solution and one of its particular solution:

$$\begin{aligned} Q(x,\overline{s}) =& Q_p + Q_g, \\ =& e^{-\sqrt{\frac{s^\alpha}{\upsilon}}x}\int_1^x -\frac{e^{\zeta\sqrt{\frac{s^\alpha}{\upsilon}}}f(\zeta)}{2\sqrt{\frac{s^\alpha}{\upsilon}}}d\zeta + e^{\sqrt{\frac{s^\alpha}{\upsilon}}x}\int_1^x \frac{e^{-\xi\sqrt{\frac{s^\alpha}{\upsilon}}}f(\xi)}{2\sqrt{\frac{s^\alpha}{\upsilon}}}d\xi \\ &+ c_1 e^{\sqrt{\frac{s^\alpha}{\upsilon}}x} + c_2 e^{-\sqrt{\frac{s^\alpha}{\upsilon}}x}, \end{aligned} \tag{4.3}$$

which is rearranged to be:

$$\begin{aligned} Q(x,\overline{s}) =& \frac{\sqrt{\frac{s^\alpha}{\upsilon}}}{2s}e^{-x\sqrt{\frac{s^\alpha}{\upsilon}}}\int_1^x e^{\zeta\sqrt{\frac{s^\alpha}{\upsilon}}+\frac{\cos\pi\zeta-1}{2\pi\upsilon}}d\zeta - \frac{\sqrt{\frac{s^\alpha}{\upsilon}}}{2s}e^{x\sqrt{\frac{s^\alpha}{\upsilon}}}\int_1^x e^{-\xi\sqrt{\frac{s^\alpha}{\upsilon}}+\frac{\cos\pi\xi-1}{2\pi\upsilon}}d\xi \\ &+ c_1 e^{\sqrt{\frac{s^\alpha}{\upsilon}}x} + c_2 e^{-\sqrt{\frac{s^\alpha}{\upsilon}}x}. \end{aligned} \tag{4.4}$$

The first order derivative of the Laplace transformed solution $Q_x(x,\overline{s})$ is calculated as:

$$\begin{aligned} Q_x(x,\overline{s}) =& -\frac{s^\alpha}{2s\upsilon}e^{-x\sqrt{\frac{s^\alpha}{\upsilon}}}\int_1^x e^{\xi\sqrt{\frac{s^\alpha}{\upsilon}}+\frac{\cos\pi\xi-1}{2\pi\upsilon}}d\xi - \frac{s^\alpha}{2s\upsilon}e^{x\sqrt{\frac{s^\alpha}{\upsilon}}}\int_1^x e^{-\zeta\sqrt{\frac{s^\alpha}{\upsilon}}+\frac{\cos\pi\zeta-1}{2\pi\upsilon}}d\zeta \\ &+ c_1\sqrt{\frac{s^\alpha}{\upsilon}}e^{\sqrt{\frac{s^\alpha}{\upsilon}}x} - c_2\sqrt{\frac{s^\alpha}{\upsilon}}e^{-\sqrt{\frac{s^\alpha}{\upsilon}}x}. \end{aligned} \tag{4.5}$$

To get the constant c_1 and c_2, we perform Laplace transform on the boundary conditions(2.8) and put them back into the equation (4.4) to calculate c_1 and c_2:

$$\begin{cases} \mathscr{L}\{q_x(0,t)\} = 0, \\ \mathscr{L}\{q_x(1,t)\} = 0, \end{cases}$$

based on which the initial condition yields to:

$$\begin{cases} Q_x(1,s) = c_1 e^{\sqrt{\frac{s^\alpha}{\upsilon}}} - c_2 e^{-\sqrt{\frac{s^\alpha}{\upsilon}}} = 0, \\ Q_x(0,s) = \sqrt{\frac{s^\alpha}{\upsilon}}c_1 - \sqrt{\frac{s^\alpha}{\upsilon}}c_2 - \frac{s^\alpha}{2s\upsilon}\int_1^0 e^{\xi\sqrt{\frac{s^\alpha}{\upsilon}}+\frac{\cos\pi\xi-1}{2\pi\upsilon}}d\xi \\ \qquad - \frac{s^\alpha}{2s\upsilon}\int_1^0 e^{-\zeta\sqrt{\frac{s^\alpha}{\upsilon}}+\frac{\cos\pi\zeta-1}{2\pi\upsilon}}d\zeta = 0, \end{cases} \tag{4.6}$$

and c_1, c_2 are calculated as:

$$\begin{cases} c_1 = \dfrac{s^\alpha}{2s\upsilon\sqrt{\frac{s^\alpha}{\upsilon}}(1-e^{2\sqrt{\frac{s^\alpha}{\upsilon}}})} \left(\displaystyle\int_1^0 e^{\xi\sqrt{\frac{s^\alpha}{\upsilon}} + \frac{\cos\pi\xi - 1}{2\pi\upsilon}} d\xi + \int_1^0 e^{-\zeta\sqrt{\frac{s^\alpha}{\upsilon}} + \frac{\cos(\pi\zeta)-1}{2\pi\upsilon}} d\zeta \right), \\ c_2 = \dfrac{s^\alpha e^{2\sqrt{\frac{s^\alpha}{\upsilon}}}}{2s\upsilon\sqrt{\frac{s^\alpha}{\upsilon}}(1-e^{2\sqrt{\frac{s^\alpha}{\upsilon}}})} \left(\displaystyle\int_1^0 e^{\xi\sqrt{\frac{s^\alpha}{\upsilon}} + \frac{\cos(\pi\xi) - 1}{2\pi\upsilon}} d\xi + \int_1^0 e^{-\zeta\sqrt{\frac{s^\alpha}{\upsilon}} + \frac{\cos\pi\zeta-1}{2\pi\upsilon}} d\zeta \right). \end{cases} \tag{4.7}$$

Since the left-hand side of the equation to be verified corresponds to the general solution of the original equation, and since $Q(x,0)$ represents a point on $Q(\overline{x},t)$ when $t=0$ (or equivalently, $s=0$), there must exist a pair of constants c_1, c_2 that satisfies the equation.

Remark 4.1. We can also find the a discretized form of $Q(x,s)$ with the finite difference method according to equation (4.4):

$$\frac{Q(x_{i+1},s_j) - 2Q(x_i,s_j) + Q(x_{i-1},s_j)}{\Delta x^2} - s_j^\alpha Q(x_i,s_j) = -s_j^{\alpha-1} e^{-\frac{1-\cos(\pi x_i)}{2\pi}},$$

where $Q_{ij} = Q(x_i,s_j)$.

4.2. Inverse Laplace transform

With the given function $F(s)$ in equation (3.1), the inverse Laplace transform can be defined as follows:

$$\mathscr{L}^{-1}(F(s)) = f(t).$$

In the context of inverting the Laplace transform, certain functions can be readily addressed using established tables. However, in our case, the Laplace transform result of $Q(s)$ does not correspond to any known formulation. As a result, we apply numerical approximation methods to compute the inverse Laplace-transformed $q(x,t)$. Among the widely recognized methods are the Gaver-Stehfest method [24], Schapery's method [45], Möbius transformation methods [2,20], Talbot method [9], and Fourier series method [14,21].

In this paper, we employ the Gaver-Stehfest approximation to reconstruct the original function $q(x,t)$ from $Q(x,s)$. The approximation is given by the following expression:

$$q(\overline{x},t)|_N \approx \frac{\ln 2}{t} \sum_{k=1}^{N} V_k Q\left(\overline{x}, k\frac{\ln 2}{t}\right), \tag{4.8}$$

where V_k represents the coefficients associated with the Gaver-tehfest method:

$$V_k = (-1)^{k+N/2} \sum_{j=\lfloor (k+1)/2 \rfloor}^{\min(k,N/2)} \frac{j^{\frac{N}{2}}(2j)!}{(\frac{N}{2}-j)!j!(j-1)!(k-j)!(2j-k)!}. \tag{4.9}$$

For a fixed value of x, the function $q(x,t)$ is approximated as a linear combination of $Q(x,s)$. The parameter N must be chosen larger than the number of decimal digits of precision. In our simulations, we set $N=12$. The approximation error is dependent on N, and a rigorous error analysis will be conducted in future work.

4.3. Inverse Cole-Hopf transform

Upon obtaining the value of $q(x,t)$, a restoration to the initial function $u(x,t)$ is facilitated through equation (2.6) as follows:

$$u(x,t) = -2\upsilon \frac{q_x(x,t)}{q(x,t)}, \tag{4.10}$$

where $q_x(x,t)$ can be approximated by the finite difference method and the numerical examples following show that.

5. Numerical examples

Using CHT method [37], the exact solution of fractional Burgers' equation is:

$$u(x,t) = 2\upsilon\pi \left[\frac{\sum_{n=1}^{\infty} a_n e^{-\frac{n^2\pi^2\upsilon t^\alpha}{\alpha}} n \sin(n\pi x)}{a_0 + \sum_{n=1}^{\infty} a_n e^{-\frac{n^2\pi^2\upsilon t^\alpha}{\alpha}} n \cos(n\pi x)} \right], \tag{5.1}$$

where

$$a_0 = \int_0^1 e^{-(\frac{1-\cos(\pi x)}{2\upsilon\pi})} dx,$$

and

$$a_n = 2\int_0^1 e^{-(\frac{1-\cos(\pi x)}{2\upsilon\pi})} \cos(n\pi x) dx.$$

To compute the numerical approximation, we utilize the gradient function from Numpy in Python to perform the finite difference for dx. The gradient is calculated using second-order accurate central differences in the interior, and second differences are applied at the boundaries.

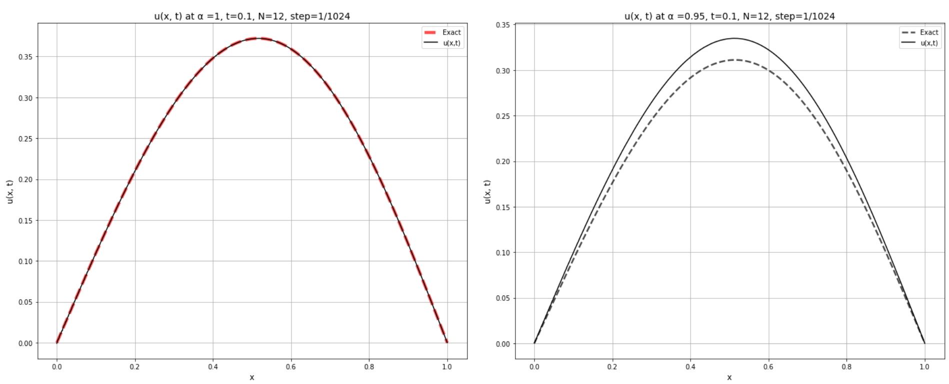
Example 5.1. To evaluate the accuracy and efficiency of the model, we set $\alpha = 1$ and compute the numerical solution at $t = 0.1$ and $t = 0.2$. The step size for dx is chosen as 1/1024.

Table 1. Absolute Error at $\alpha = 1$ for different t values

x	$t = 0.1$			$t = 0.2$		
	Numerical	**Exact**	**Abs. Error**	**Numerical**	**Exact**	**Abs. Error**
0.1	0.10930	0.10911	1.90E-04	0.04178	0.04177	4.81E-06
0.2	0.20904	0.20906	1.90E-05	0.07974	0.07972	2.21E-05
0.3	0.29173	0.29177	3.49E-05	0.11060	0.11057	3.02E-05
0.4	0.34770	0.34772	1.97E-05	0.13079	0.13074	5.07E-05
0.5	0.37156	0.37162	5.87E-05	0.13853	0.13847	5.22E-05
0.6	0.35916	0.35920	4.67E-05	0.13269	0.13263	5.54E-05
0.7	0.31042	0.31044	1.93E-05	0.11372	0.11367	5.10E-05
0.8	0.22800	0.22799	3.92E-06	0.08295	0.08291	3.91E-05
0.9	0.12136	0.12135	1.02E-05	0.04396	0.04394	2.14E-05

Table 1 shows that our model reaches good stability and accuracy at integer order. The absolute error compared with corresponding exact solution is below 0.0001 which is better than the results in [22].

The plot of $u(x,t)$ for $\alpha = 1$ at $t = 0.1$ shows that even with a small step size of $dx = 1/1000$, the curve is almost indistinguishable from the exact solution, confirming the accuracy of the method. The total computation time for solving 1000 points is approximately 3.06 seconds, with each step taking around 3 milliseconds—an impressively fast result compared to existing methods.



(a) $u(x,t)$ at $\alpha = 1$, $t = 0.1$ and step=1/1024 (b) $u(x,t)$ at $t = 0.1, \alpha = 0.95$ and step=1/1024

Figure 1. $u(x,s)$ at different α

Furthermore, we measured the computation time for various step sizes and compared these results with those obtained using the CHT method [37]. The table below illustrates the computational efficiency of our proposed method in comparison to the CHT approach.

Table 2. Comparison of time consumption for different time steps between two methods. *Note: The computations were performed using a single-core CPU (M1 Pro).*

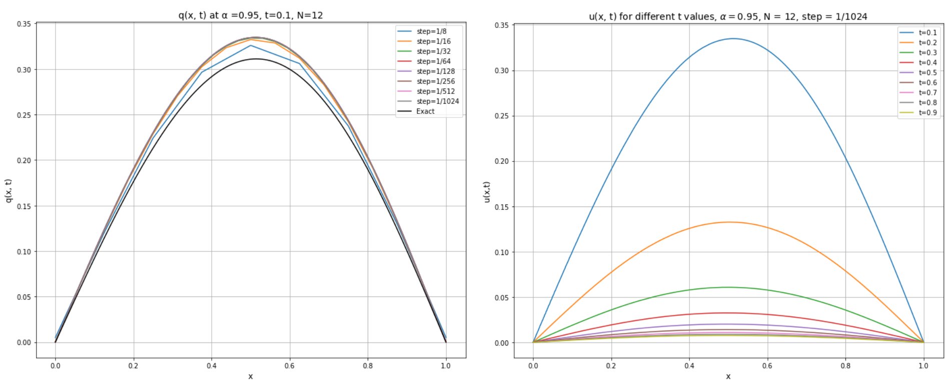
Step	Our Method		Method in [37]	
	Total Time (s)	Time per Step (s)	Total Time (s)	Time per Step (s)
1/12	0.0468	0.0036	3.0012	0.2309
1/64	0.2043	0.0031	11.1696	0.1718
1/128	0.3961	0.0031	22.4827	0.1743
1/256	0.7802	0.0030	43.8430	0.1706
1/512	1.5637	0.0030	89.8373	0.1751
1/1024	3.1223	0.0030	178.1930	0.1738

Example 5.2. To testify the stability and accuracy of the model in fractional orders, we calculate $u(x,t)$ at different α values.

Table 3. Absolute Error for different α values at $t = 0.1$

x	$\alpha = 0.99$			$\alpha = 0.95$			$\alpha = 0.9$		
	Numerical	**Exact**	**Abs. Error**	**Numerical**	**Exact**	**Abs. Error**	**Numerical**	**Exact**	**Abs. Error**
0.1	0.1071	0.1057	0.0014	0.0993	0.0918	0.0076	0.0905	0.0745	0.0160
0.2	0.2050	0.2024	0.0026	0.1898	0.1757	0.0142	0.1728	0.1425	0.0303
0.3	0.2860	0.2824	0.0036	0.2643	0.2449	0.0194	0.2400	0.1983	0.0417
0.4	0.3407	0.3365	0.0042	0.3140	0.2913	0.0227	0.2844	0.2355	0.0489
0.5	0.3638	0.3595	0.0043	0.3345	0.3108	0.0237	0.3020	0.2507	0.0513
0.6	0.3514	0.3473	0.0041	0.3222	0.2998	0.0224	0.2900	0.2413	0.0487
0.7	0.3036	0.3001	0.0035	0.2777	0.2586	0.0190	0.2491	0.2077	0.0414
0.8	0.2228	0.2203	0.0025	0.2034	0.1897	0.0138	0.1821	0.1521	0.0301
0.9	0.1186	0.1173	0.0013	0.1081	0.1009	0.0073	0.0966	0.0808	0.0159

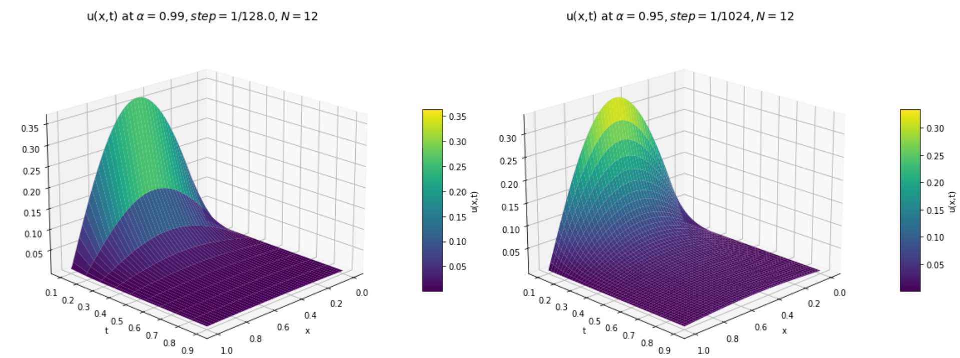
And we plot the lines of $u(x, t)$ in different t and see how the solution evolves with the t dimension. As supposed, it keeps in the shape of sin function in the half interval but the amplitude decreases as t increases.



(a) $u(x, t)$ with different step of dx (b) $u(x, t)$ for group of t at $\alpha = 0.95$

Figure 2. $u(x, s)$ at different x(left) or t(right)

The gradual and smooth progression of the curves also reflects the model's stability. As a result, we proceed to a detailed visualization of the 3D surface of $u(x, t)$ (see Figure 3). The surface plot, showing $u(x, t)$ for $t \in (0, 1)$ and $x \in (0, 1)$, demonstrates a smooth gradient, consistent with the Mean Absolute Error (MAE) analysis mentioned earlier. This 3D representation provides a comprehensive view of the initial solutions to the fractional Burgers' equation in the bi-dimensional space of x and t, highlighting favourable characteristics of the solution.



(a) $\alpha = 0.99$ (b) $\alpha = 0.95$

Figure 3. 3-D graph of $u(x,t)$ at different α

In addition, we analyse the absolute error from a 3D perspective. Compared with the scale of absolute errors displayed in [1], our model continues to perform well under these conditions.

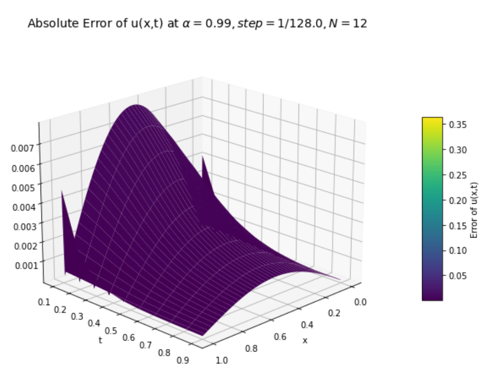


Figure 4. Absolute error of $u(x,t)$ at $\alpha = 0.99$

Example 5.3. In the third example, we examine the model with smaller values of α, specifically setting $\alpha = 0.5$. We plot the surface of the solution in the same manner as before, and observe that the surface maintains a good resemblance to the exact solution described in equation (5.1). At this stage, however, the error becomes more pronounced.

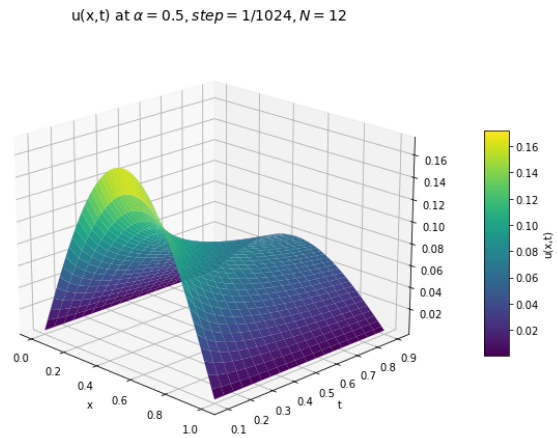


Figure 5. Surface for $u(x,t)$ at $\alpha = 0.5$.

6. Convergence and stability analysis

We compute the Mean Absolute Error (MAE) of the solution using various step sizes for dx to assess the order of convergence. Using the solution with a step size of $1/1024$ as the benchmark for the best available approximation, we observe that as the step size decreases, the MAE also reduces to an ideal range. Additionally, the reduction in MAE from step $= 1/8$ to step $= 1/1024$ indicates the model's stability. As the step size decreases beyond $1/1024$, the MAE stabilizes, likely due to the limitations in the approximation of $q(x,t)$ in equation (4.8).

Table 4. MAE Analysis of Example when $t = 0.1, \alpha = 0.95$

x \ step	1/8	1/16	1/32	1/64	1/128	1/256	1/512	1/1024
0.1	0.0049	0.0623	0.0933	0.0935	0.0935	0.0973	0.0992	0.0993
0.2	0.1202	0.1787	0.1795	0.1798	0.1865	0.1898	0.1898	0.1898
0.3	0.2240	0.2285	0.2517	0.2623	0.2624	0.2624	0.2636	0.2642
0.4	0.2965	0.3024	0.3038	0.3106	0.3136	0.3136	0.3136	0.3140
0.5	0.3259	0.3323	0.3339	0.3343	0.3344	0.3344	0.3344	0.3344
0.6	0.3259	0.3285	0.3233	0.3237	0.3238	0.3227	0.3222	0.3222
0.7	0.3059	0.2827	0.2840	0.2843	0.2799	0.2776	0.2776	0.2776
0.8	0.2374	0.2419	0.2187	0.2058	0.2059	0.2059	0.2042	0.2034
0.9	0.1297	0.1321	0.1327	0.1171	0.1091	0.1091	0.1091	0.1081
MAE	0.0944	0.0385	0.0246	0.0100	0.0058	0.0025	0.0010	

To further verify the stability of the solution for small α values, we analyse the incremental behaviour of $u(x,t)$ as dx increases with a step size of $1/1024$. The increments of $u(x,t)$ exhibit smooth behaviour, indicating that the model remains stable at each successive step.

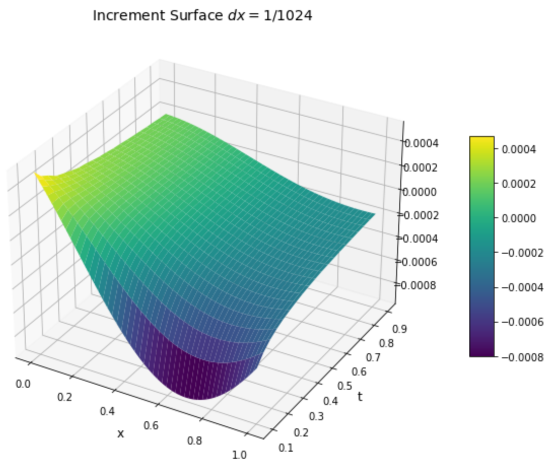


Figure 6. 1/1024 Increment of $u(x,t)$ at $\alpha = 0.5$.

7. Conclusion

In this paper, we investigate the fractional Burgers' equation using the Caputo derivative. The Laplace transform method is employed to convert the problem into an ordinary differential equation, which features a complex structure. Since an analytical solution is only available for specific fractional ODEs, we apply a numerical method to compute the inverse Laplace transform. The numerical examples demonstrate a high level of accuracy, with results reaching approximately 99%. Although minor deviations are present at certain points, they remain within acceptable bounds. Additionally, the proposed method proves to be highly efficient, requiring less than 1/50 of the computational time compared to the CHT method. Future research will explore the application of this method to higher-dimensional cases, given its potential for both feasibility and accuracy.

References

[1] T. Akram, M. Abbas, M. B. Riaz et al., *An efficient numerical technique for solving time fractional Burgers equation*, Alexandria Engineering Journal, 2020, 59(4), 2201–2220.

[2] M. Asif, S. Mairaj, Z. Saeed et al., *A novel image encryption technique based on mobius transformation*, Computational Intelligence and Neuroscience, 2021, 2021.

[3] A. R. Bahadır, *A fully implicit finite-difference scheme for two-dimensional Burgers' equations*, Applied Mathematics and Computation, 2003, 137(1), 131–137.

[4] M. P. Bonkile, A. Awasthi, C. Lakshmi et al., *A systematic literature review of Burgers' equation with recent advances*, Pramana, 2018, 90, 1–21.

[5] M. Caputo, *Linear Models of Dissipation whose Q is almost Frequency Independent—II*, Geophysical Journal International, 1967, 13(5), 529–539.

[6] S. Cengizci and Ö. Uğur, *A stabilized fem formulation with discontinuity-capturing for solving Burgers' -type equations at high reynolds numbers*, Applied Mathematics and Computation, 2023, 442, 127705.

[7] C.-M. Chen, F. Liu, I. Turner and V. Anh, *A fourier method for the fractional diffusion equation describing sub-diffusion*, Journal of Computational Physics, 2007, 227(2), 886–897.

[8] S. Dhawan, S. Kapoor, S. Kumar and S. Rawat, *Contemporary review of techniques for the solution of nonlinear Burgers equation*, Journal of Computational Science, 2012, 3(5), 405–419.

[9] B. Dingfelder and J. Weideman, *An improved talbot method for numerical Laplace transform inversion*, Numerical Algorithms, 2015, 68(1), 167–183.

[10] G.-h. Gao, Z.-z. Sun and H.-w. Zhang, *A new fractional numerical differentiation formula to approximate the caputo fractional derivative and its applications*, Journal of Computational Physics, 2014, 259, 33–50.

[11] A. Gorguis, *A comparison between Cole–Hopf transformation and the decomposition method for solving Burgers' equations*, Applied Mathematics and Computation, 2006, 173(1), 126–136.

[12] L. Guo, H. Wu, X. Yu and T. Zhou, *Monte carlo fpinns: Deep learning method for forward and inverse problems involving high dimensional fractional partial differential equations*, Computer Methods in Applied Mechanics and Engineering, 2022, 400, 115523.

[13] X. Hu and L. Zhang, *An analysis of a second order difference scheme for the fractional subdiffusion system*, Applied Mathematical Modelling, 2016, 40(2), 1634–1649.

[14] S. Ichikawa and A. Kishima, *Applications of fourier series technique to inverse Laplace transform*, Memoirs of the Faculty of Engineering, Kyoto University, 1972, 34(1), 53–67.

[15] M. Izadi and H. Srivastava, *A novel matrix technique for multi-order pantograph differential equations of fractional order*, Proceedings of the Royal Society A, 2021, 477(2253), 20210321.

[16] H. Jassim, J. Vahidi and V. Ariyan, *Solving Laplace equation within local fractional operators by using local fractional differential transform and Laplace variational iteration methods*, Nonlinear Dynamics and Systems Theory, 2020, 20(4), 388–396.

[17] Y. Jiang and J. Ma, *High-order finite element methods for time-fractional partial differential equations*, Journal of Computational and Applied Mathematics, 2011, 235(11), 3285–3290.

[18] R. Jiwari, *A hybrid numerical scheme for the numerical solution of the Burgers' equation*, Computer Physics Communications, 2015, 188, 59–67.

[19] R. Kannan and Z. Wang, *A high order spectral volume solution to the Burgers' equation using the hopf–cole transformation*, International journal for numerical methods in fluids, 2012, 69(4), 781–801.

[20] V. G. Kim, Y. Lipman, X. Chen and T. Funkhouser, *Möbius transformations for global intrinsic symmetry analysis*, in *Computer Graphics Forum*, 29, Wiley Online Library, 2010, 1689–1700.

[21] K. L. Kuhlman, *Review of inverse Laplace transform algorithms for Laplace-space numerical approaches*, Numerical Algorithms, 2013, 63, 339–355.

[22] A. Kurt, Y. Çenesiz and O. Tasbozan, *On the solution of Burgers' equation with the new fractional derivative*, Open Physics, 2015, 13(1).

[23] S. Kutluay, A. Bahadir and A. Özdeş, *Numerical solution of one-dimensional Burgers' equation: explicit and exact-explicit finite difference methods*, Journal of computational and applied mathematics, 1999, 103(2), 251–261.

[24] A. Kuznetsov, *On the convergence of the Gaver-Stehfest algorithm*, SIAM Journal on Numerical Analysis, 2013, 51(6), 2984–2998.

[25] T. Langlands and B. I. Henry, *The accuracy and stability of an implicit solution method for the fractional diffusion equation*, Journal of Computational Physics, 2005, 205(2), 719–736.

[26] H. Li et al., *State estimation for fractional-order complex dynamical networks with linear fractional parametric uncertainty*, in *Abstract and Applied Analysis*, 2013, Hindawi, 2013.

[27] M. Li, D. Shi and L. Pei, *Convergence and superconvergence analysis of finite element methods for the time fractional diffusion equation*, Applied Numerical Mathematics, 2020, 151, 141–160.

[28] Y. Li and N. Sun, *Numerical solution of fractional differential equations using the generalized block pulse operational matrix*, Computers & Mathematics with Applications, 2011, 62(3), 1046–1054.

[29] H. Liu, Y. Fu, B. Li et al., *Discrete waveform relaxation method for linear fractional delay differential-algebraic equations*, Discrete Dynamics in Nature and Society, 2017, 2017.

[30] L. Lu, X. Meng, Z. Mao and G. E. Karniadakis, *Deepxde: A deep learning library for solving differential equations*, SIAM review, 2021, 63(1), 208–228.

[31] Y. Luchko and J. Trujillo, *Caputo-type modification of the erdélyi-kober fractional derivative*, Fractional Calculus and Applied Analysis, 2007, 10(3), 249–267.

[32] A. Majeed, M. Kamran and M. Rafique, *An approximation to the solution of time fractional modified Burgers' equation using extended cubic b-spline method*, Computational and Applied Mathematics, 2020, 39(4), 257.

[33] K. S. Miller and B. Ross, *An introduction to the fractional calculus and fractional differential equations*, 1993.

[34] M. Nadeem, J.-H. He and A. Islam, *The homotopy perturbation method for fractional differential equations*, International Journal of Numerical Methods for Heat & Fluid Flow, 2021, 31(11), 3490–3504.

[35] T. F. Nonnenmacher and R. Metzler, *On the Riemann-Liouville fractional calculus and some recent applications*, Fractals, 1995, 3(03), 557–566.

[36] Z. M. Odibat, *A study on the convergence of variational iteration method*, Mathematical and Computer Modelling, 2010, 51(9-10), 1181–1192.

[37] T. Ohwada, *Cole-Hopf transformation as numerical tool for the Burgers' equation*, Appl. Comput. Math, 2009, 8(1), 107–113.

[38] M. D. Ortigueira and J. T. Machado, *What is a fractional derivative?*, Journal of Computational Physics, 2015, 293, 4–13.

[39] E. Rothe, *Zweidimensionale parabolische randwertaufgaben als grenzfall eindimensionaler randwertaufgaben*, Mathematische Annalen, 1930, 102(1), 650–670.

[40] P. Roul, V. P. Goura and R. Cavoretto, *A numerical technique based on b-spline for a class of time-fractional diffusion equation*, Numerical Methods for Partial Differential Equations, 2023, 39(1), 45–64.

[41] P. Roul, V. P. Goura, H. Madduri and K. Obaidurrahman, *Design and stability analysis of an implicit non-standard finite difference scheme for fractional neutron point kinetic equation*, Applied Numerical Mathematics, 2019, 145, 201–226.

[42] P. Roul, V. Rohil, G. Espinosa-Paredes et al., *Design and analysis of a numerical method for fractional neutron diffusion equation with delayed neutrons*, Applied Numerical Mathematics, 2020, 157, 634–653.

[43] P. Sachdev, *A generalised Cole-Hopf transformation for nonlinear parabolic and hyperbolic equations*, Zeitschrift für angewandte Mathematik und Physik ZAMP, 1978, 29, 963–970.

[44] A. H. Salas, C. A. Gómez S et al., *Application of the Cole-Hopf transformation for finding exact solutions to several forms of the seventh-order KdV equation*, Mathematical Problems in Engineering, 2010, 2010.

[45] R. A. Schapery, *A method of viscoelastic stress analysis using elastic solutions*, Journal of the Franklin Institute, 1965, 279(4), 268–289.

[46] B. K. Singh and M. Gupta, *A new efficient fourth order collocation scheme for solving Burgers' equation*, Applied Mathematics and Computation, 2021, 399, 126011.

[47] J. Singh, *Analysis of fractional blood alcohol model with composite fractional derivative*, Chaos, Solitons & Fractals, 2020, 140, 110127.

[48] V. K. Srivastava, M. K. Awasthi and S. Singh, *An implicit logarithmic finite-difference technique for two dimensional coupled viscous Burgers' equation*, Aip Advances, 2013, 3(12).

[49] J.-L. Wu, *A wavelet operational method for solving fractional partial differential equations numerically*, Applied Mathematics and Computation, 2009, 214(1), 31–40.

[50] T. Yan, *The numerical solutions for the nonhomogeneous Burgers' equation with the generalized hopf-cole transformation.*, Networks & Heterogeneous Media, 2023, 18(1).

[51] X.-J. Yang, J. T. Machado and J. Hristov, *Nonlinear dynamics for local fractional Burgers' equation arising in fractal flow*, Nonlinear Dynamics, 2016, 84, 3–7.

[52] L. Zada and I. Aziz, *Numerical solution of fractional partial differential equations via Haar wavelet*, Numerical Methods for Partial Differential Equations, 2022, 38(2), 222–242.