Journal of Applied Analysis and Computation
Volume 15, Number 6, December 2025, 3226–3245
Website:http://www.jaac-online.com
DOI:10.11948/20240527

A LINEAR TIME-SPACE HIGH-ORDER COMPACT FINITE DIFFERENCE SCHEME FOR SOLVING THE ONE-DIMENSIONAL NONLINEAR BURGERS' EQUATION*

Xueqing Miao[1], Jiaye Gan[2], Tingfu Ma[1] and Lili Wu[1,†]

Abstract This paper presents a novel finite difference scheme for solving the one-dimensional (1D) nonlinear Burgers' equation with high accuracy. A fourth-order backward difference formula is employed to discretize the time derivative, while the nonlinear residual terms are efficiently linearized via Taylor series expansion. Spatial discretization is achieved using a fourth-order compact difference scheme for second-order derivatives and a fourth-order Padé scheme for first-order derivatives. As a result, the proposed approach yields a linear compact finite difference scheme with fourth-order accuracy in both temporal and spatial dimensions. The accuracy and stability of this method are demonstrated through a series of numerical experiments, validating its effectiveness for solving nonlinear Burgers' equations. The high-order finite difference scheme for the nonlinear Burgers' equation can accurately simulate phenomena in fields such as physics, sound wave propagation, and aerodynamics.

Keywords 1D nonlinear Burgers' equation, compact finite difference scheme, linear scheme, time-space high accuracy.

MSC(2010) 65M12, 65M06, 35Q75.

1. Introduction

The Burgers' equation, introduced by American physicist John von Neumann Burgers [3] in 1948, has emerged as a pivotal model in the study of turbulence. Renowned for its simplicity and ability to encapsulate essential phenomena in fluid dynamics, the equation has since become a cornerstone in diverse scientific and engineering applications. Its utility spans computational fluid dynamics, nonlinear optics, and

†The corresponding author.
[1]School of Mathematics and Computer Science, Ningxia Normal University, Guyuan 756000, China
[2]School of Power and Mechanical Engineering, Wuhan University, Wuhan 430072, China
*This work is partially supported by National Natural Science Foundation of China (12261070), Foreign Expert Program (G2023045002L), Natural Science Foundation of Ningxia (2022AAC03314), the Key Research and Development Program of Ningxia (2021BEB04053), Research Startup Funding for Full-time High-level Talents Introduced by Ningxia Hui Autonomous Region (2024BEH04096).
Email: 3132616437@qq.com(X. Miao), gjywhu@hotmail.com(J. Gan), tingfma@nxnu.edu.cn(T. Ma), wll82003019@nxnu.edu.cn(L. Wu)

communication computing [5, 7, 13, 30, 38], underscoring its versatility and significance in advancing both theoretical and applied research.

In recent years, substantial advancements have been achieved in numerical methods for solving Burgers' equation. For example, Khoshfetrat et al. [15] introduced a meshless approach combining the traditional Differential Quadrature (DQ) method with the Local Multiquadric Radial Basis Function Differential Quadrature (LMQRBF-DQ), yielding a highly efficient and accurate technique suitable for complex, irregular geometries. Fan and Li [6] discretized space and time using the generalized finite difference method and implicit Euler method, respectively, and applied a fictitious time integration method to process nonlinear algebraic equations, resulting in a meshless numerical scheme for solving the two-dimensional (2D) Burgers' equation. Similarly, Maryam et al. [21] proposed a local reproducing kernel method that uses a Newton basis function space, offering flexibility in addressing mixed boundary conditions. The cubic B-spline method [2, 12, 16, 23, 26, 32, 36] has also seen widespread application in Burgers' equation solutions, with studies demonstrating its efficacy across various conditions. Dahy and Elgindy [4] presented a Cole-Hopf barycentric Gegenbauer integral pseudospectral method (CHBGPM) to enhance computational efficiency. In another approach, Lai and Ma [18] introduced a lattice Boltzmann model for coupled nonlinear viscous Burgers' equations. This approach employs a dual evolution equation, a third-order Total Variation Diminishing (TVD) Runge-Kutta method, and a central difference scheme, achieving second-order convergence in time and third-order convergence in space.

Further innovations include Shyaman et al. [33] and Tsai et al. [35] use of a finite point method, offering a convergence order of $O(\tau + h^2)$. Kutluay and Ucar [17] proposed a Galerkin quadratic B-spline finite element method, while Gao et al. [9] utilized quadratic quasi-interpolation and first-order forward finite differences for Burgers' equation. Scholars have also advanced differential moment methods [14, 24, 25], further expanding solution techniques. Ganaie et al. [8] applied a cubic Hermite collocation method, and Srivastava [34] proposed an exponential implicit numerical scheme for solving the 2D coupled Burgers' equation and verified it through numerical experimentation. Zhu et al. [42] explored the discrete Adomian decomposition method (ADM) for solving the 2D Burgers' equation, while Mukundan et al. [27] used a method of lines (MOL) approach with implicit finite difference integration to create a semi-discrete scheme for the 1D Burgers' equation. Li [20] explored the space-time generalized finite difference method (GFDM) was combined with Newton's method to stably and accurately solve 2D unsteady Burgers' equations. Abdullah et al. [1] applied an efficient computational technique for the numerical solution of the Burgers' equation is presented. The derivative in space is approximated using cubic B-splines and the Hermite formula whereas time discretization is performed by finite differences. Shallal et al. [31] applied a finite element method (FEM) with cubic Hermite element is presented to acquire the numerical solution for the 1D Burgers' equation. Palav et al. [28] applied an efficient numerical scheme for solving Burgers' equation has been developed using the uniform hyperbolic polynomial (UHP) B-spline collocation method.

The above-mentioned methods are all low-order schemes. These methods often fail to accurately capture the details of the flow field, resulting in deviations from reality in the computed results. Especially in cases with large gradients and high Reynolds numbers, where the flow field changes are more drastic and complex, low-order schemes are more prone to errors. To address this issue, high-order schemes have attracted widespread attention from researchers. For example, Pandey et al. [29] employed the Douglas finite difference scheme to simplify the Burgers' equation into a heat equation, constructing a scheme with a convergence order of $O(\tau^2 + h^4)$. Zhanlav et al. [41] proposed an explicit finite difference scheme for the numerical solution of the Burgers' equation with Robin boundary conditions. This scheme has third-order accuracy in the time direction and sixth-order accuracy in the spatial direction. Hammada and El-Azab [10] proposed a new linear numerical scheme with 2N order accuracy in space (specifically, fourth-order accuracy when N=2) and a first-order backward Euler method in time. Recent high-order approaches include Yang et al. [39,40] linear compact difference schemes, achieving a convergence order of $O(\tau^2 + h^4)$, and Hussain [11] hybrid radial basis function (HRBF) method. By applying the RK4 and SSP-RK3 methods for temporal discretization, Hussain developed high-accuracy numerical schemes for spatial operators in Burgers' equation. Finally, Mittal [22] presented a space-time pseudospectral method based on Lagrange polynomials, employing Chebyshev-Gauss-Lobatto (CGL) points to reduce the problem to a nonlinear algebraic system, efficiently solved through the Newton-Raphson method.

Existing high-order methods often fail to achieve consistent accuracy in the temporal direction, despite their high accuracy in the spatial domain. This imbalance forces the use of very small time steps to meet desired spatial accuracy, significantly increasing computation time, memory usage, and storage requirements. These demands reduce computational efficiency and raise computational costs, posing significant challenges, especially in large-scale simulations or real-time applications. Although decreasing the time step may help reduce truncation errors, excessively small time steps can amplify the effects of rounding errors, leading to distorted results or non-physical phenomena such as numerical oscillations. For solving nonlinear partial differential equations, linearization techniques offer distinct advantages in computational efficiency. Consequently, it is crucial to construct high-order compact numerical methods with equal accuracy for space-time. Such methods enhance the accuracy of numerical simulations, improve computational efficiency, and contribute substantially to advancements in numerical analysis for complex systems.

In this paper, a high accuracy compact finite difference scheme for solving 1D Burgers' equation is constructed. The first-order time derivative is discretized using a fourth-order backward difference formula, while the second-order spatial derivative is approximated with a fourth-order compact finite difference scheme. Additionally, the first-order spatial derivative is discretized via a fourth-order Padé scheme. By applying a linearization strategy, we construct a linear high-order compact finite difference scheme that achieves fourth-order accuracy in both spatial and temporal domains.

The structure of this paper is as follows: in Section 2, we introduce the equation model along with definitions for relevant operators, inner products, and norms. Section 3 presents the construction of the proposed linear high-order compact finite difference scheme in both space and time. Numerical experiments are conducted in Section 4 to demonstrate the accuracy and efficiency of the method. Finally, Section 5 states the conclusions of this study.

2. Model equation

This study aims to explore the 1D nonlinear Burgers' equation as outlined below

$$u_t + uu_x = \varepsilon u_{xx}, \quad (x,t) \in D \times [T_0, T]. \tag{2.1}$$

In this context, $u(x,t)$ represents the fluid velocity, with the nonlinear term uu_x represents the convection term, describing the convection transport process of the fluid, while u_{xx} represents the diffusion term, reflecting the diffusion of the fluid due to viscous effects. The kinematic viscosity $\varepsilon = \frac{1}{Re} > 0$ is the dimensionless number that balances the relationship between the nonlinear convection term and the viscous diffusion term, and is called the kinematic viscosity coefficient. Re is the Reynolds number, $D = \{x : a \le x \le b\}$, where a and b are contants. In addition, the initial condition related with Eq. (2.1) is given as

$$u(x, T_0) = \varphi(x), \quad x \in D, \tag{2.2}$$

and the boundary condition is as follows

$$u(x,t) = \psi(x,t), \quad (x,t) \in \partial D \times [T_0, T]. \tag{2.3}$$

We first divide the space domain $[a,b]$ by an N mesh, where $N \in \mathbf{Z}^+$, and define $h = (b-a)/N$, it is the step sizes in the spatial x direction. Let $\tau = (T - T_0)/M$ represent the time step size, where $M \in \mathbf{Z}^+$. Mesh point is marked as (x_i, t_n), $x_i = a + ih,\ t_n = n\tau,\ i = 0, 1, ..., N, 0 \le n \le M$. Let $u_i^n = u(x_i, t_n)$, U_i^n be the approximation of u_i^n. We further denote $(u_x)_i^n$ and $(u_{xx})_i^n$ for u_x and u_{xx} at grid point (x_i, t_n), respectively. $V_h = \{u_i : i = 0, 1, 2, \cdots, N\}$. For a mesh function $u \in V_h$, we define the difference operators as follows:

$$\delta_x u_i^n = \frac{u_{i+1}^n - u_{i-1}^n}{2h}, \quad \delta_x u_{i-\frac{1}{2}}^n = \frac{u_i^n - u_{i-1}^n}{h}, \quad \delta_x^2 u_i^n = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2}.$$

The fourth-order compact finite difference scheme for approximating the second-order derivative proposed as follows:

$$(u_{xx})_i = \left(1 + \frac{h^2}{12}\delta_x^2\right)^{-1} \delta_x^2 u_i + O(h^4).$$

Further, for a mesh function $u, v \in V_h$, we introduce the inner product, norm and a notation:

$$|\ v\ |^2 = h\sum_{i=1}^{N-1}\left(\delta_x v_{i-\frac{1}{2}}\right)^2, (u,v) = h\left(\frac{1}{2}u_0 v_0 + \sum_{i=1}^{N-1} u_i v_i + \frac{1}{2}u_N v_N\right), \|v\| = \sqrt{(v,v)}.$$

3. Linear high-order compact difference scheme

Consider the value of Eq. (2.1) at the point (x_i, t_{n+1}), we have

$$(u_t)_i^{n+1} + u_i^{n+1}(u_x)_i^{n+1} = \varepsilon (u_{xx})_i^{n+1}, \tag{3.1}$$

the first derivative of Eq. (3.1) with respect to time adopts a five-layer fourth-order backward difference formula

$$(u_t)_i^{n+1} = \frac{25u_i^{n+1} - 48u_i^n + 36u_i^{n-1} - 16u_i^{n-2} + 3u_i^{n-3}}{12\tau} + O(\tau^4), \tag{3.2}$$

substituting Eq. (3.2) into Eq. (3.1), we have

$$\begin{aligned}&\frac{25u_i^{n+1}-48u_i^n+36u_i^{n-1}-16u_i^{n-2}+3u_i^{n-3}}{12\tau}+u_i^{n+1}(u_x)_i^{n+1}\\&=\varepsilon(u_{xx})_i^{n+1}+O(\tau^4).\end{aligned}\tag{3.3}$$

The fourth-order compact difference scheme is used for the second-order derivative of space in Eq. (3.3), we can get

$$\begin{aligned}&\frac{25u_i^{n+1}-48u_i^n+36u_i^{n-1}-16u_i^{n-2}+3u_i^{n-3}}{12\tau}+u_i^{n+1}(u_x)_i^{n+1}\\&=\varepsilon\left(1+\frac{h^2}{12}\delta_x^2\right)^{-1}\delta_x^2u_i^{n+1}+O(\tau^4+h^4),\end{aligned}\tag{3.4}$$

multiply both sides of Eq. (3.4) by $(1+\frac{h^2}{12}\delta_x^2)$, and arrange the result

$$\begin{aligned}&25\left(1+\frac{h^2}{12}\delta_x^2\right)u_i^{n+1}-12\tau\varepsilon\delta_x^2u_i^{n+1}+12\tau\left(1+\frac{h^2}{12}\delta_x^2\right)u_i^{n+1}(u_x)_i^{n+1}\\&=-\left(1+\frac{h^2}{12}\delta_x^2\right)\left(-48u_i^n+36u_i^{n-1}-16u_i^{n-2}+3u_i^{n-3}\right)+O(\tau^4+h^4),\end{aligned}\tag{3.5}$$

substitute the definition of the central difference operator into the Eq. (3.5), sort out data, we can get

$$\begin{aligned}&\left[\frac{125}{6}+\frac{24\tau\varepsilon}{h^2}+10(u_x)_i^{n+1}\tau\right]u_i^{n+1}+\left[\frac{25}{12}-\frac{12\tau\varepsilon}{h^2}+(u_x)_{i+1}^{n+1}\tau\right]u_{i+1}^{n+1}\\&\quad+\left[\frac{25}{12}-\frac{12\tau\varepsilon}{h^2}+(u_x)_{i-1}^{n+1}\tau\right]u_{i-1}^{n+1}\\&=40u_i^n+4(u_{i+1}^n+u_{i-1}^n)\\&\quad-30u_i^{n-1}-3(u_{i+1}^{n-1}+u_{i-1}^{n-1})-\frac{40}{3}u_i^{n-2}+\frac{4}{3}(u_{i+1}^{n-2}+u_{i-1}^{n-2})\\&\quad-\frac{5}{2}u_i^{n-3}-\frac{1}{4}(u_{i+1}^{n-3}+u_{i-1}^{n-3})+O(\tau^4+h^4).\end{aligned}\tag{3.6}$$

To linearize the $(n+1)$ layer nonlinear first-order derivative term $(u_x)_i^{n+1}$, setting $g_i=(u_x)_i$, using the Taylor series expansion method has,

$$g_i^{n+1}=g_i^n+\tau(g_t)_i^n+\frac{\tau^2}{2}(g_{tt})_i^n+\frac{\tau^3}{6}(g_{ttt})_i^n+O(\tau^4),\tag{3.7}$$

$$g_i^{n-1}=g_i^n-\tau(g_t)_i^n+\frac{\tau^2}{2}(g_{tt})_i^n-\frac{\tau^3}{6}(g_{ttt})_i^n+O(\tau^4),\tag{3.8}$$

$$g_i^{n-2}=g_i^n-2\tau(g_t)_i^n+\frac{(2\tau)^2}{2}(g_{tt})_i^n-\frac{(2\tau)^3}{6}(g_{ttt})_i^n+O(\tau^4),\tag{3.9}$$

$$g_i^{n-3}=g_i^n-3\tau(g_t)_i^n+\frac{(3\tau)^2}{2}(g_{tt})_i^n-\frac{(3\tau)^3}{6}(g_{ttt})_i^n+O(\tau^4),\tag{3.10}$$

let (3.7)+α (3.8)+β (3.9)+γ (3.10), we have

$$g_i^{n+1}+\alpha g_i^{n-1}+\beta g_i^{n-2}+\gamma g_i^{n-3}$$

$$
\begin{aligned}
=&(1+\alpha+\beta+\gamma)g_i^n+\tau(1-\alpha-2\beta-3\gamma)(g_t)_i^n\\
&+\frac{\tau^2}{2}(1+\alpha+4\beta+9\gamma)(g_{tt})_i^n+\frac{\tau^3}{6}(1-\alpha-8\beta-27\gamma)(g_{ttt})_i^n+O(\tau^4),
\end{aligned}\tag{3.11}
$$

command $\begin{cases}1-\alpha-2\beta-3\gamma=0\\1+\alpha+4\beta+9\gamma=0\\1-\alpha-8\beta-27\gamma=0\end{cases}$, solve that $\begin{cases}\alpha=6\\\beta=-4\\\gamma=1\end{cases}$, substituting $\alpha,\ \beta,\ \gamma$ into Eq. (3.11), we have

$$
g_i^{n+1}=4g_i^n-6g_i^{n-1}+4g_i^{n-2}-g_i^{n-3}+O(\tau^4),\tag{3.12}
$$

that is

$$
(u_x)_i^{n+1}=4(u_x)_i^n-6(u_x)_i^{n-1}+4(u_x)_i^{n-2}-(u_x)_i^{n-3}+O(\tau^4),\tag{3.13}
$$

let $f_i=(u_x)_i^{n+1}$, substituting Eq. (3.13) into Eq. (3.6) and omitting the truncation error term, we have

$$
\begin{aligned}
&\left(\frac{125}{6}+\frac{24\tau\varepsilon}{h^2}+10\tau f_i\right)U_i^{n+1}+\left(\frac{25}{12}-\frac{12\tau\varepsilon}{h^2}+\tau f_{i+1}\right)U_{i+1}^{n+1}\\
&+(\frac{25}{12}-\frac{12\tau\varepsilon}{h^2}+\tau f_{i-1})U_{i-1}^{n+1}\\
=&40U_i^n+4(U_{i+1}^n+U_{i-1}^n)-30U_i^{n-1}-3(U_{i+1}^{n-1}+U_{i-1}^{n-1})-\frac{40}{3}U_i^{n-2}\\
&+\frac{4}{3}(U_{i+1}^{n-2}+U_{i-1}^{n-2})-\frac{5}{2}U_i^{n-3}-\frac{1}{4}(U_{i+1}^{n-3}+U_{i-1}^{n-3}),
\end{aligned}\tag{3.14}
$$

where U_i^n is the numerical solution of u_i^n.

The first derivative in the spatial direction in Eq. (3.14) can be computed by the fourth-order Padé scheme [19] as

$$
\frac{1}{6}(u_x)_{i+1}+\frac{2}{3}(u_x)_i+\frac{1}{6}(u_x)_{i-1}=\frac{u_{i+1}-u_{i-1}}{2h}+O(h^4),\tag{3.15}
$$

where $i=1,2,\cdots,N-1$.

Lemma 3.1. (See [37]) *If we set $N\geq 5$, then there exists $\xi_k(0\leq k\leq 4)$ such that*

$$
\frac{2}{3}(u_x)_0+\frac{1}{3}(u_x)_1=\sum_{k=0}^{4}\xi_k\delta_x u_{k+\frac{1}{2}}+O(h^4),\tag{3.16}
$$

$$
\frac{1}{3}(u_x)_{N-1}+\frac{2}{3}(u_x)_N=\sum_{k=0}^{4}\xi_k\delta_x u_{N-\frac{1}{2}-k}+O(h^4).\tag{3.17}
$$

Thus, the fourth-order Padé scheme (3.15) with boundary conditions (3.16) and (3.17) has the following properties:

$$
\parallel u_x\parallel\leq 3\sqrt{1+\sum_{k=0}^{4}\xi_k^2}\mid u\mid+3\parallel r\parallel .\tag{3.18}
$$

Where r is the remainder of the truncation error. The values of ξ_j $(j = 0, 1, 2, 3, 4)$ in Eqs. (3.16) and (3.17) can be obtained using Taylor series expansion, taking $\xi_0 = \frac{143}{90}$, $\xi_1 = -\frac{83}{60}$, $\xi_2 = \frac{77}{60}$, $\xi_3 = -\frac{109}{180}$, $\xi_4 = \frac{7}{60}$, and in the x-directions at the boundary points for the first-order derivative, a fourth-order boundary schemes can be obtained:

$$\begin{aligned}\frac{2}{3}(u_x)_0 + \frac{1}{3}(u_x)_1 &= \frac{1}{h}\left[-\frac{143}{90}u_0 + \frac{107}{36}u_1 - \frac{8}{3}u_2\right]\\ &\quad + \frac{1}{h}\left[\frac{17}{9}u_3 - \frac{13}{18}u_4 + \frac{7}{60}u_5\right] + O(h^4),\end{aligned} \tag{3.19}$$

$$\begin{aligned}\frac{1}{3}(u_x)_{N-1} + \frac{2}{3}(u_x)_N &= \frac{1}{h}\left[\frac{143}{90}u_N - \frac{107}{36}u_{N-1} + \frac{8}{3}u_{N-2}\right]\\ &\quad + \frac{1}{h}\left[-\frac{17}{9}u_{N-3} + \frac{13}{18}u_{N-4} - \frac{7}{60}u_{N-5}\right] + O(h^4).\end{aligned} \tag{3.20}$$

Thus, we obtain a five-layer linear high-order compact (LHOC) difference scheme for solving Eqs. (2.1)–(2.3) with both fourth-order accuracy in both the temporal and spatial directions. Since this is a five-layer scheme, a startup scheme is necessary to calculate the values of u^1, u^2, and u^3.

Consider the value of Eq. (2.1) at point $(x_i, t_{n+\frac{1}{2}})$, we have

$$(u_t)_i^{n+\frac{1}{2}} + (uu_x)_i^{n+\frac{1}{2}} = \varepsilon(u_{xx})_i^{n+\frac{1}{2}}, \tag{3.21}$$

in the above formula, the Crank-Nicolson scheme is used to discrete the time direction

$$\frac{u_i^{n+1} - u_i^n}{\tau} + \frac{1}{2}\left[(uu_x)_i^{n+1} + (uu_x)_i^n\right] = \frac{\varepsilon}{2}[(u_{xx})_i^{n+1} + (u_{xx})_i^n] + O(\tau^2), \tag{3.22}$$

the fourth-order compact difference scheme is used for the function containing the second-order derivative in Eq. (3.22),

$$\begin{aligned}&\frac{u_i^{n+1} - u_i^n}{\tau} + \frac{1}{2}\left[(uu_x)_i^{n+1} + (uu_x)_i^n\right]\\ &= \frac{\varepsilon}{2}\left(1 + \frac{h^2}{12}\delta_x^2\right)^{-1}\left[\delta_x^2 u_i^{n+1} + \delta_x^2 u_i^n\right] + O(\tau^2 + h^4),\end{aligned} \tag{3.23}$$

sorting out Eq. (3.23), let's multiply both sides of this equation by $\left(1 + \frac{h^2}{12}\delta_x^2\right)$

$$\begin{aligned}&\left(1 + \frac{h^2}{12}\delta_x^2\right)\frac{u_i^{n+1} - u_i^n}{\tau} + \frac{1}{2}\left(1 + \frac{h^2}{12}\delta_x^2\right)\left[(uu_x)_i^{n+1} + (uu_x)_i^n\right]\\ &= \frac{\varepsilon}{2}\left[\delta_x^2 u_i^{n+1} + \delta_x^2 u_i^n\right] + O(\tau^2 + h^4).\end{aligned} \tag{3.24}$$

The first-order derivative of the spatial direction in Eq. (3.23) can be obtained by using Eqs. (3.15), (3.19), and (3.20). To ensure that the startup scheme achieves the same fourth-order temporal accuracy as the main scheme (3.14), we apply the Richardson extrapolation technique. By using the following extrapolation formula, we enhance the time accuracy of the startup scheme (3.24) to fourth-order, aligning it with the temporal accuracy of the primary scheme

$$\hat{U}_i^n(h,\tau) = \frac{1}{3}\left[4U_i^{2n}(h, \frac{\tau}{2}) - U_i^n(h,\tau)\right], \quad 0 \le i \le N, \quad 1 \le n \le 3. \tag{3.25}$$

The $\hat{U}_i^n(h,\tau)$ in the above equation represents the extrapolation solution on the $(n)th$ time layer. $U_i^n(h,\tau)$ and $U_i^{2n}(h,\frac{\tau}{2})$ represent the solution calculated by the startup scheme (3.24) at layer n when the time step is τ and the time step is $\frac{\tau}{2}$, respectively.

4. Numerical experiment

In this section, we present several numerical experiments to evaluate the performance and various properties of the proposed scheme in solving the system described by Eqs. (2.1)–(2.3). The L_∞ and L_2 norm errors are defined as:

$$L_\infty = \underset{i}{Max}\,|U_i - u_i|\,, \qquad L_2 = \sqrt{h\sum_{i=1}^{N-1}(U_i-u_i)^2}.$$

The temporal and spatial convergence rate are calculated as:

$$\text{Rate}_\tau = \frac{\ln\left[\text{error}(\tau_1)/\text{error}(\tau_2)\right]}{\ln(\tau_1/\tau_2)}, \qquad \text{Rate}_h = \frac{\ln\left[\text{error}(h_1)/\text{error}(h_2)\right]}{\ln(h_1/h_2)},$$

where $error(\tau_1)$ and $error(\tau_2)$ are the L_∞ norm errors with grid sizes of τ_1 and τ_2, $error(h_1)$ and $error(h_2)$ are the L_∞ norm errors with grid sizes of h_1 and h_2, respectively.

Example 4.1. Consider the 1D Burgers' Eq. (2.1) at the following initial and boundary conditions:

$$u(x,0) = \frac{2\varepsilon\pi\sin(\pi x)}{\sigma+\cos(\pi x)}, \quad 0 \le x \le 1,$$
$$u(0,t) = u(1,t) = 0, \qquad 0 \le t \le T.$$

The exact solution to the example is as follows:

$$u(x,t) = \frac{2\varepsilon\pi\exp(-\pi^2\varepsilon t)\sin(\pi x)}{\sigma+\exp(-\pi^2\varepsilon t)\cos(\pi x)}.$$

Tables 1–5 present the L_2 and L_∞ norm error values for both the proposed LHOC scheme and various methods from the literature, tested under different parameters. The data demonstrate that the proposed method achieves higher accuracy than existing methods. Figure 1 illustrates the behavior of the solution layers for Example 4.1 as affected by the viscosity parameter ε. Even as ε approaches zero and layer formation intensifies, the LHOC scheme maintains an almost exact alignment with the analytical solutions. Furthermore, as ε approaches zero, the peaks of the error plots in Figure 2 shift towards the right side of the domain, corresponding to the influence of the boundary layer. Despite this shift, the LHOC scheme effectively manages the error bounds, keeping them within infinitesimal limits and demonstrating robust performance near boundary layers.

Table 6 compares the L_∞ norm errors and convergence rates between the method from Ref. [27] and the proposed LHOC scheme for Example 4.1. As the grid is refined, the LHOC scheme demonstrates a more rapid decrease in errors, highlighting

Table 1. Comparison of the L_2 and L_∞ norm errors of Example 4.1 of various methods at $\sigma = 2$, $h = 0.1$ and $T = 2$ for various τ and ε.

τ	ε	Ref. [33]		LHOC	
		L_2	L_∞	L_2	L_∞
0.1	$1.00E-01$	$6.666E-03$	$1.112E-02$	$9.695E-06$	$1.388E-05$
	$1.00E-02$	$1.435E-04$	$2.052E-04$	$6.389E-06$	$1.591E-05$
	$1.00E-03$	$2.491E-06$	$5.336E-06$	$2.773E-07$	$8.514E-07$
	$1.00E-04$	$2.868E-08$	$6.540E-08$	$3.630E-09$	$1.137E-08$
0.0001	$1.00E-01$	$7.828E-04$	$1.120E-03$	$5.821E-06$	$8.756E-06$
	$1.00E-02$	$1.323E-04$	$2.761E-04$	$6.100E-06$	$1.553E-05$
	$1.00E-03$	$2.646E-06$	$5.875E-06$	$2.454E-07$	$7.353E-07$
	$1.00E-04$	$2.886E-08$	$6.602E-08$	$3.024E-09$	$9.207E-09$

Table 2. Comparison of the L_2 and L_∞ norm errors for Example 4.1 of various methods at $\sigma = 2$, $\tau = 0.001$ and $T = 1$ for various h and ε.

h	ε	Ref. [33]		LHOC	
		L_2	L_∞	L_2	L_∞
0.1	$1.00E-01$	$1.34E-03$	$2.07E-03$	$1.711E-05$	$3.237E-05$
	$1.00E-02$	$9.37E-05$	$1.97E-04$	$5.953E-06$	$1.654E-05$
	$1.00E-03$	$1.39E-06$	$3.13E-06$	$1.378E-07$	$4.169E-07$
	$1.00E-04$	$1.45E-08$	$3.32E-08$	$1.536E-09$	$4.686E-09$
0.0125	$1.00E-01$	$1.09E-03$	$1.75E-03$	$3.647E-09$	$7.084E-09$
	$1.00E-02$	$6.12E-06$	$9.92E-06$	$1.268E-09$	$3.478E-09$
	$1.00E-03$	$1.74E-08$	$3.87E-08$	$2.981E-11$	$8.987E-11$
	$1.00E-04$	$2.30E-10$	$5.67E-10$	$3.340E-13$	$1.026E-12$

Table 3. Comparison of the L_2 norm errors for Example 4.1 of various methods at $\sigma = 2$, $\varepsilon = 1$, $h = 1/40$ and $\tau = 0.0001$ and different T.

T	Ref. [29]	Ref. [9]	Ref. [35]	Ref. [33]	LHOC
0.2	$3.14E-04$	$2.16E-04$	$4.74E-05$	$2.99E-03$	$2.005E-07$
0.4	$4.35E-05$	$2.77E-04$	$8.24E-06$	$5.15E-04$	$3.981E-08$
0.6	$6.04E-06$	$3.74E-04$	$1.38E-06$	$7.97E-05$	$7.396E-09$
0.8	$8.39E-07$	$4.05E-04$	$2.52E-07$	$1.20E-05$	$1.285E-09$
1.0	$1.17E-07$	$5.19E-04$	$5.39E-08$	$1.81E-06$	$2.145E-10$

Table 4. Comparison of the L_∞ errors for Example 4.1 of various methods at $T = 1$ with $\sigma = 2$, $\tau = 0.0001$, $h = 1/20$ and different ε.

ε	Ref. [25]	Ref. [11]	Ref. [11]	Ref. [33]	LHOC
$1.00E-01$	$1.5E-03$	$2.04E-04$	$3.42E-05$	$5.11E-04$	$1.861E-06$
$1.00E-02$	$1.6E-04$	$7.32E-06$	$7.90E-06$	$5.32E-05$	$9.226E-07$
$1.00E-03$	$3.3E-06$	$1.20E-07$	$3.06E-07$	$8.63E-07$	$2.424E-08$
$1.00E-04$	$2.7E-08$	$1.64E-09$	$3.17E-09$	$9.10E-10$	$3.151E-10$
$1.00E-05$	$5.5E-10$	$1.69E-11$	$3.12E-11$	$9.15E-11$	$3.252E-12$

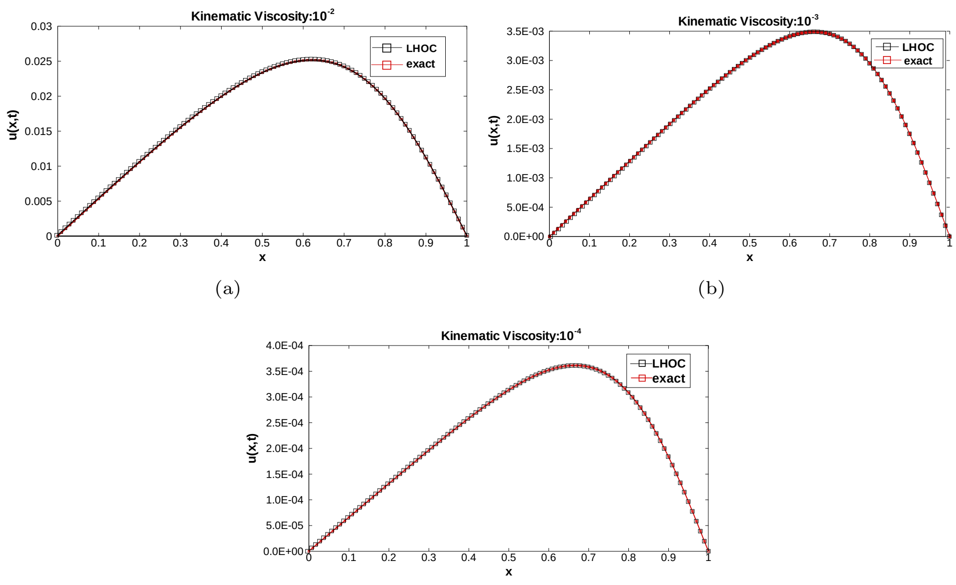


Figure 1. The exact solution (red) and the numerical solution (black) of Example 4.1 obtained using the LHOC scheme for various values of kinematic viscosity (a) $\varepsilon = 0.01$ (b) $\varepsilon = 0.001$ (c) $\varepsilon = 0.0001$ with $h = 0.01$, $\tau = 0.001$ and $T = 3$.

Table 5. Comparison of the L_2 norm errors for Example 4.1 of various methods at $N = 18$ and $N = 40$ with $\sigma = 2,\ \tau = 0.0001,\ T = 0.001$ at different ε.

ε	Ref. [8] (N=18)	Ref. [23] (N=40)	LHOC (N=18)	LHOC (N=40)
0.5	$3.54E-05$	$2.79E-05$	$3.649E-06$	$9.642E-08$
0.2	$5.24E-07$	$4.57E-06$	$6.523E-07$	$1.593E-08$
0.1	$3.54E-07$	$1.15E-06$	$1.704E-07$	$4.027E-09$

Table 6. Comparison of the L_∞ norm errors and convergence rate for Example 4.1 of various methods at $\sigma = 100,\ \tau = 0.01,\ T = 1$, and $\varepsilon = 0.005$ for various N.

N	Ref. [27]		LHOC	
	L_∞	$Rate_h$	L_∞	$Rate_h$
10	$1.209E-07$	–	$6.717E-10$	–
20	$3.046E-08$	1.99	$4.204E-11$	3.99
40	$8.672E-09$	1.81	$2.630E-12$	4.00
80	$5.078E-09$	0.77	$1.643E-13$	4.00

its superior convergence efficiency. Additionally, the LHOC scheme consistently achieves a convergence rate close to 4 across all grid refinements, underscoring its stability and high accuracy. Figure 3 provides graphical representations of the numerical solutions obtained with the LHOC scheme under various Reynolds Re numbers and time T values, illustrating its performance across different parameter settings.

Example 4.2. Consider the 1D Burgers' Eq. (2.1) at the following initial and

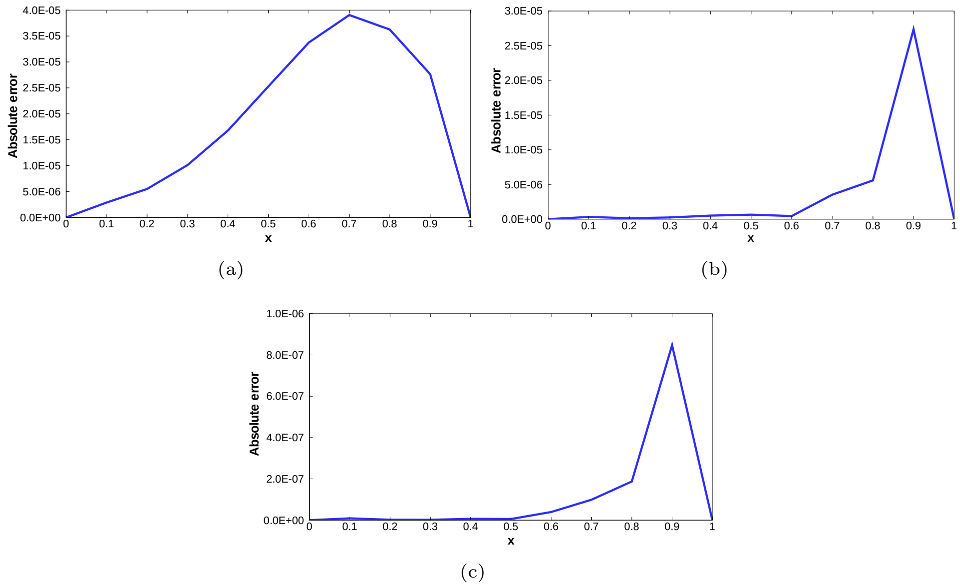


Figure 2. Absolute errors of Example 4.1 at $T = 1$ with $h = 0.1$, $\tau = 0.01$, and (a) $\varepsilon = 0.1$ (b) $\varepsilon = 0.01$ (c) $\varepsilon = 0.001$.

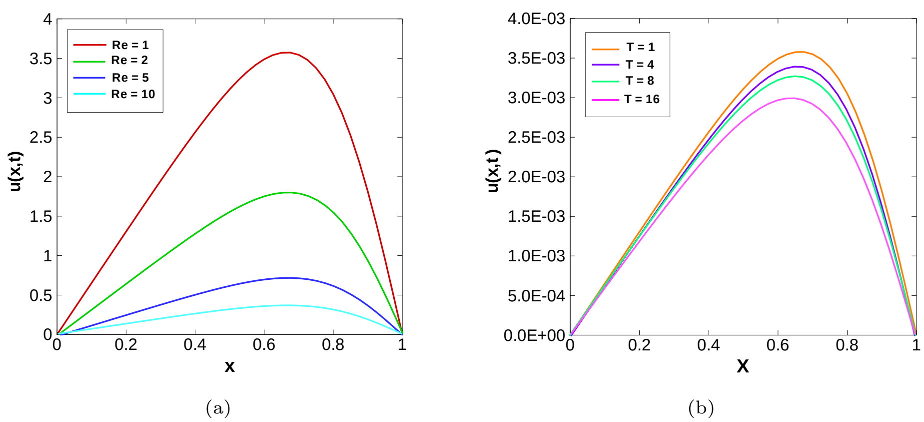


Figure 3. (a) the numerical solution of the LHOC scheme when $Re = 1/\varepsilon = 1,\ 2,\ 5,\ 10$, $N = 40$, $T = 0.001$, $\tau = 0.0001$, (b) the numerical solution of the LHOC scheme when $T = 1,\ 4,\ 8,\ 16$, $N = 40$, $\varepsilon = 0.001$, $\tau = 0.01$ for Example 4.1.

boundary conditions:

$$u(x,1) = \frac{x}{1 + \exp\left[\frac{1}{4\varepsilon}\left(x^2 - \frac{1}{4}\right)\right]}, \quad 1 \le t \le T,$$

$$u(a,t) = \frac{\frac{a}{t}}{1 + \left[\frac{t}{\exp\left(\frac{1}{8\varepsilon}\right)}\right]^{\frac{1}{2}} \exp\left(\frac{a^2}{4\varepsilon t}\right)}, \ u(b,t) = \frac{\frac{b}{t}}{1 + \left[\frac{t}{\exp\left(\frac{1}{8\varepsilon}\right)}\right]^{\frac{1}{2}} \exp\left(\frac{b^2}{4\varepsilon t}\right)}, \ a \le x \le b.$$

Table 7. Comparison of the L_2 and L_∞ norm errors for Example 4.2 of various methods at $\tau = 0.001,\ \varepsilon = 0.005,\ h = 0.005$ and $a = 0,\ b = 1.2$ for various T.

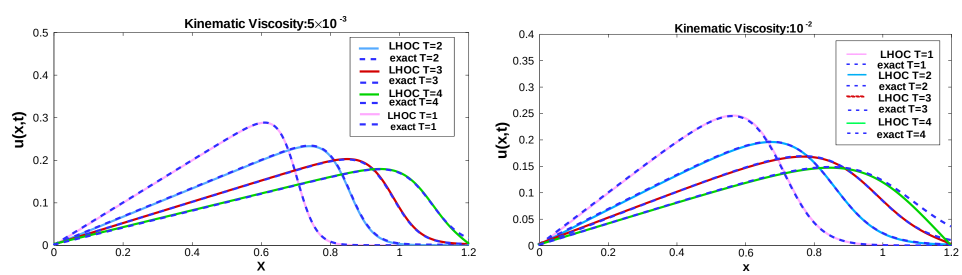
T	Ref. [33]		LHOC	
	L_2	L_∞	L_2	L_∞
1.7	$1.21E-03$	$5.16E-03$	$4.945E-08$	$1.921E-07$
2.5	$1.33E-03$	$5.09E-03$	$7.591E-06$	$5.754E-05$
3	$1.36E-03$	$4.93E-03$	$8.586E-05$	$6.143E-04$
3.5	$1.38E-03$	$4.78E-03$	$5.537E-04$	$3.755E-03$

Table 8. Comparison of the L_2 and L_∞ norm errors for Example 4.2 of various methods at $\tau = 0.001,\ T = 3.6,\ \varepsilon = 0.005$ and $a = 0,\ b = 1$ for various N.

N	Ref. [23]		Ref. [40]		LHOC	
	L_2	L_∞	L_2	L_∞	L_2	L_∞
40	$3.1E-04$	$1.0E-03$	$4.44E-04$	$1.49E-03$	$3.139E-06$	$1.422E-05$
60	$1.5E-04$	$4.7E-04$	$9.20E-05$	$3.07E-04$	$6.871E-07$	$3.221E-06$
80	$0.8E-04$	$2.7E-04$	$2.98E-05$	$9.96E-05$	$2.202E-07$	$9.849E-07$
100	$0.6E-04$	$1.7E-04$	$1.25E-05$	$4.16E-05$	$9.010E-08$	$4.106E-07$
120	$0.4E-04$	$1.2E-04$	$6.18E-06$	$2.04E-05$	$4.333E-08$	$1.966E-07$

Table 9. Comparison of the L_2 and L_∞ norm errors for Example 4.2 of various methods at $\tau = 0.001,\ \varepsilon = 0.005,\ h = 0.005$ and $a = 0,\ b = 1$ for various T.

T	Ref. [23]		Ref. [40]		LHOC	
	L_2	L_∞	L_2	L_∞	L_2	L_∞
1.7	$2.52E-05$	$9.94E-05$	$8.04E-07$	$2.56E-06$	$4.388E-08$	$1.649E-07$
2.5	$1.51E-05$	$5.49E-05$	$9.72E-07$	$3.15E-06$	$2.428E-08$	$8.450E-08$
3.0	$1.18E-05$	$4.14E-05$	$9.90E-07$	$3.12E-06$	$1.447E-08$	$6.021E-08$
3.5	$1.17E-05$	$4.86E-05$	$9.87E-07$	$3.02E-06$	$8.077E-09$	$3.626E-08$



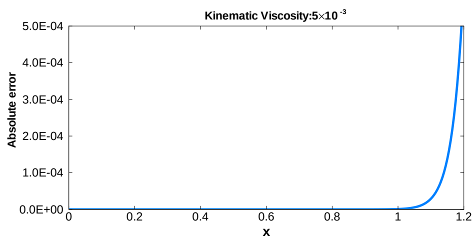
(a) (b)

Figure 4. The exact solution and the numerical solution of Example 4.2 obtained using the LHOC scheme with $h = 0.005$, $\tau = 0.001$ at $T = 1,\ 2,\ 3,\ 4$ for various values of kinematic viscosity (a) $\varepsilon = 0.005$ (b) $\varepsilon = 0.01$.

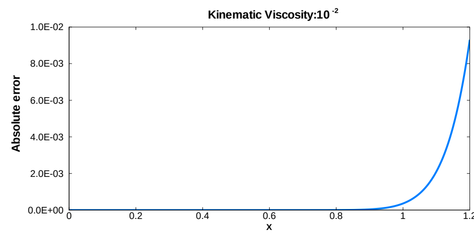
The exact solution to the example is as follows:

$$u(x,t) = \frac{\frac{x}{t}}{1 + \left[\frac{t}{\exp\left(\frac{1}{8\varepsilon}\right)}\right]^{\frac{1}{2}} \exp\left(\frac{x^2}{4\varepsilon t}\right)}.$$

Tables 7–9 present the error comparisons between the proposed LHOC method and the reference methods. Among all the methods evaluated, the LHOC method consistently produces the smallest errors. Figure 4 shows the superposition of the LHOC numerical solution for Example 4.2 over its exact solution, highlighting the excellent agreement between the two. Additionally, the error plots in Figure 5 indicate that the error bounds of the LHOC solutions remain within acceptable limits, further demonstrating the method's accuracy and reliability.



(a)



(b)

Figure 5. Absolute errors of Example 4.2 with $h = 0.005$, $\tau = 0.001$ at $T = 3$ and (a) $\varepsilon = 0.005$ (b) $\varepsilon = 0.01$.

Example 4.3. The Coupled Burgers' Equation (CBE) equation is given in the following form:

$$
\begin{aligned}
&u_t - u_{xx} + k_1 u u_x + k_2 (uv)_x = 0, \\
&v_t - v_{xx} + k_1 v v_x + k_3 (uv)_x = 0.
\end{aligned}
$$

Consider the CBE for $k_1 = -2$ and $k_2 = k_3 = 1$ at the following initial and boundary conditions:

$$
\begin{aligned}
&u(x,0) = \sin(x), \quad v(x,0) = \sin(x), \quad x \in [-\pi, \pi], \\
&u(-\pi,t) = u(\pi,t) = v(-\pi,t) = v(\pi,t) = 0, \quad t \in [0,T].
\end{aligned}
$$

The exact solution to the example is as follows:

$$
u(x,t) = v(x,t) = \exp(-t)\sin(x).
$$

Table 10 and Table 11 show the L_2 and L_∞ norm errors under different parameters for Example 4.3, the results obtained by the presented method are better than all of the compared ones. Figure 6 shows numerical simulations of LHOC scheme of Example 4.3 for values of $N = 100$, $\tau = 0.001$, $k_1 = -2$, $k_2 = k_3 = 1$ at times $T = 1,\ 2$, and 3.

Example 4.4. Consider the CBE for $k_1 = 2$ with different values of k_2 and k_3 at the following initial conditions:

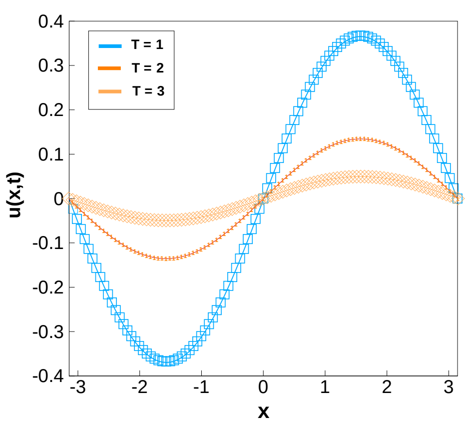
$$
\begin{aligned}
&u(x,0) = a_0 - 2\mathcal{A}\left(\frac{2k_2 - 1}{4k_2k_3 - 1}\right)\frac{\sinh(\mathcal{A}x)}{\cosh(\mathcal{A}x)}, \\
&v(x,0) = a_0\left(\frac{2k_3 - 1}{2k_2 - 1}\right) - 2\mathcal{A}\left(\frac{2k_2 - 1}{4k_2k_3 - 1}\right)\frac{\sinh(\mathcal{A}x)}{\cosh(\mathcal{A}x)}.
\end{aligned}
$$

Table 10. Comparison of the calculated L_2 and L_∞ norm errors of Example 4.3 with $\tau = 0.01$ at different T on $[-\pi, \pi]$.

N	T	Ref. [9]		Ref. [36]		LHOC	
		L_2	L_∞	L_2	L_∞	L_2	L_∞
50	0.1	$1.227E-04$	$1.108E-04$	$1.004E-06$	$9.075E-07$	$2.114E-06$	$1.247E-06$
	0.5	$6.136E-04$	$3.714E-04$	$5.000E-06$	$3.038E-06$	$4.204E-06$	$2.545E-06$
	1.0	$1.228E-03$	$4.507E-04$	$9.955E-06$	$3.671E-06$	$2.240E-06$	$1.466E-06$
	1.5	$1.842E-03$	$4.102E-04$	$1.486E-05$	$3.325E-06$	$1.052E-06$	$7.382E-07$
	2.0	$2.457E-03$	$3.318E-04$	$1.973E-05$	$2.677E-06$	$5.821E-07$	$3.972E-07$
	2.5	$3.072E-03$	$2.517E-04$	$2.454E-05$	$2.021E-06$	$3.845E-07$	$2.397E-07$
	3.0	$3.687E-03$	$1.832E-04$	$2.931E-05$	$1.464E-06$	$2.698E-07$	$1.580E-07$
200	0.1	$6.878E-06$	$6.224E-06$	$8.338E-07$	$7.546E-07$	$2.352E-08$	$1.346E-08$
	0.5	$3.439E-05$	$2.086E-05$	$4.165E-06$	$2.528E-06$	$4.657E-08$	$2.674E-08$
	1.0	$6.878E-05$	$2.530E-05$	$8.321E-06$	$3.064E-06$	$2.399E-08$	$1.406E-08$
	1.5	$1.032E-04$	$2.302E-05$	$1.247E-06$	$2.785E-06$	$9.813E-09$	$5.986E-09$
	2.0	$1.376E-04$	$1.862E-05$	$1.660E-06$	$2.249E-06$	$3.835E-09$	$2.491E-09$
	2.5	$1.720E-04$	$1.412E-05$	$2.073E-06$	$1.703E-06$	$1.567E-09$	$1.088E-09$
	3.0	$2.064E-04$	$1.027E-05$	$2.485E-06$	$1.238E-06$	$7.418E-10$	$5.195E-10$

Table 11. Comparison of the calculated L_2 and L_∞ norm errors for Example 4.3 with $N = 200$ and 400, $\tau = 0.001$ at different T.

N	T	Ref. [17]		Ref. [36]		LHOC	
		L_2	L_∞	L_2	L_∞	L_2	L_∞
200	0.1	$0.17E-06$	$0.52E-06$	$8.997E-09$	$8.143E-09$	$1.044E-08$	$6.141E-09$
	0.5	$0.27E-06$	$0.36E-06$	$4.495E-08$	$2.728E-08$	$1.680E-08$	$1.028E-08$
	1.0	$0.36E-06$	$0.22E-06$	$8.979E-08$	$3.306E-08$	$8.808E-09$	$5.822E-09$
400	0.1	$0.07E-06$	$0.14E-06$	$8.374E-09$	$7.578E-09$	$6.542E-10$	$3.848E-10$
	0.5	$0.16E-06$	$0.14E-06$	$4.185E-08$	$2.539E-08$	$1.053E-09$	$6.438E-10$
	1.0	$0.15E-06$	$0.10E-06$	$8.365E-08$	$3.079E-08$	$5.516E-10$	$3.646E-10$



(a)

Figure 6. Numerical solution of LHOC scheme of Example 4.3 for values of $N = 100$, $\tau = 0.001$, $k_1 = -2$, $k_2 = k_3 = 1$ at times $T = 1,\ 2,$ and 3.

The exact solution to the example is as follows:

$$u(x,t) = a_0 - 2\mathcal{A}\left(\frac{2k_2 - 1}{4k_2k_3 - 1}\right)\frac{\sinh[\mathcal{A}\,(x - 2\mathcal{A}t)]}{\cosh[\mathcal{A}(x - 2\mathcal{A}t)]},$$

$$v(x,t) = a_0\left(\frac{2k_3-1}{2k_2-1}\right) - 2\mathcal{A}\left(\frac{2k_2-1}{4k_2k_3-1}\right)\frac{\sinh[\mathcal{A}(x-2\mathcal{A}t)]}{\cosh[\mathcal{A}(x-2\mathcal{A}t)]},$$

where

$$a_0 = 0.05, \quad \mathcal{A} = \frac{1}{2}\left[\frac{a_0(4k_2k_3-1)}{2k_2-1}\right], \quad -10 \le x \le 10.$$

The boundary conditions for the example are given by the exact solution. Tables

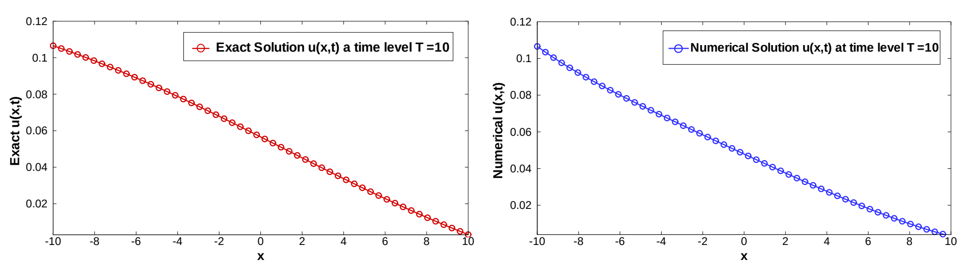
Table 12. Comparison of the L_2 norm errors for Example 4.4 of various methods with $\tau = 0.01$, $N = 50$ for $k_2 = 0.1$ and $k_3 = 0.3$ at different T.

T	Ref. [17]		Ref. [36]		LHOC	
	$error_u$	$error_v$	$error_u$	$error_v$	$error_u$	$error_v$
0.1	$5.975E-04$	$6.332E-04$	$1.489E-04$	$1.145E-04$	$3.443E-05$	$1.341E-05$
0.5	$2.911E-03$	$3.079E-03$	$7.287E-04$	$5.547E-04$	$1.566E-04$	$6.044E-05$
1.0	$5.705E-03$	$6.027E-03$	$1.432E-03$	$1.082E-03$	$3.051E-04$	$1.169E-04$
1.5	$8.420E-03$	$8.885E-03$	$2.119E-03$	$1.593E-03$	$4.503E-04$	$1.716E-04$
2.0	$1.107E-02$	$1.167E-02$	$2.792E-03$	$2.089E-03$	$5.928E-04$	$2.249E-04$
2.5	$1.366E-02$	$1.439E-02$	$3.453E-03$	$2.573E-03$	$7.330E-04$	$2.770E-04$
3.0	$1.620E-02$	$1.705E-02$	$4.103E-03$	$3.046E-03$	$8.710E-04$	$3.280E-04$

Table 13. Comparison of the L_2 norm errors for Example 4.4 of various methods with $\tau = 0.01$, $N = 100$ at different k_2, k_3 and T.

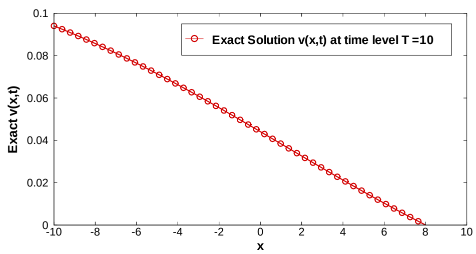
T	k_2	k_3	Ref. [17]		Ref. [36]		LHOC	
			$error_u$	$error_v$	$error_u$	$error_v$	$error_u$	$error_v$
0.5	0.1	0.30	$6.783E-04$	$5.101E-04$	$6.766E-04$	$5.044E-04$	$1.566E-04$	$6.044E-05$
	0.3	0.003	$7.609E-04$	$1.327E-03$	$7.456E-04$	$1.322E-03$	$1.716E-04$	$7.699E-04$
1	0.1	0.30	$1.334E-03$	$0.995E-03$	$1.330E-03$	$9.836E-04$	$3.051E-04$	$1.169E-04$
	0.3	0.003	$1.500E-03$	$2.617E-03$	$1.469E-03$	$2.607E-03$	$3.362E-04$	$1.506E-03$

12 and 13 present the L_2 norm errors of the proposed algorithm in comparison with other methods as the time T varies. In all scenarios, the LHOC scheme exhibits the lowest L_2 norm error, highlighting its superior numerical performance for this example compared to the methods in Refs. [17,36]. Figures 7 and 8 display the exact and numerical solutions of $u(x,t)$ under specific time and parameter conditions. A comparison of the two subplots reveals that the exact solution $u(x,t)$ closely matches the numerical solution, demonstrating the high accuracy of the LHOC scheme.

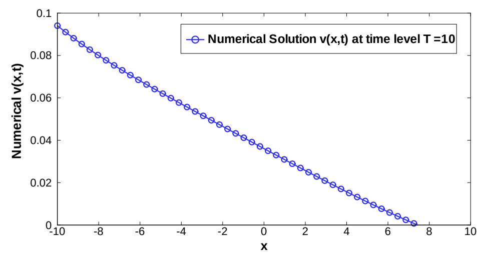


(a) (b)

Figure 7. Exact and numerical solutions of Example 4.4 for values of $a_0 = 0.05$, $k_1 = -1$, $k_2 = 0.1$, $k_3 = 0.2$, $\tau = 0.001$ and $N = 51$.



(a)



(b)

Figure 8. Exact and numerical solutions of Example 4.4 for values of $a_0 = 0.05$, $k_1 = -1$, $k_2 = 0.1$, $k_3 = 0.2$, $\tau = 0.001$ and $N = 51$.

Example 4.5. Consider CBE for $k_1 = -2$ and $k_2 = k_3 = \frac{5}{2}$ at the following initial and boundary conditions,

$$u(x,0) = v(x,0) = \lambda\left[1 - \frac{1}{\coth(\frac{3}{2}\lambda x)}\right], \quad x \in [-20, 20],$$

$$u(-20,t) = v(-20,t) = \lambda\left[1 - \frac{1}{\coth\left(\frac{3}{2}\lambda(-20 - 3\lambda t)\right)}\right],$$

$$u(20,t) = v(20,t) = \lambda\left[1 - \frac{1}{\coth\left(\frac{3}{2}\lambda(20 - 3\lambda t)\right)}\right], \quad t > 0.$$

The exact solution to the example is as follows:

$$u(x,t) = v(x,t) = \lambda\left[1 - \frac{1}{\coth\left(\frac{3}{2}\lambda(x - 3\lambda t)\right)}\right].$$

Table 14. Comparison of the calculated L_2 and L_∞ error norms for Example 4.5 with $N = 320$, $\tau = 0.001$ and $\lambda = 0.1$ at different T.

T	Ref. [18]		Ref. [14]		LHOC	
	L_2	L_∞	L_2	L_∞	L_2	L_∞
1	$1.572E-06$	$6.286E-07$	$1.934E-06$	$2.582E-07$	$3.097E-10$	$1.607E-10$
2	$2.938E-06$	$1.114E-06$	$2.495E-06$	$2.881E-07$	$3.377E-10$	$1.356E-10$
3	$4.168E-06$	$1.588E-06$	$2.977E-06$	$3.177E-07$	$4.104E-10$	$1.787E-10$
4	$5.250E-06$	$1.987E-06$	$3.444E-06$	$3.490E-07$	$4.783E-10$	$2.123E-10$
5	$6.188E-06$	$2.347E-06$	$3.920E-06$	$3.828E-07$	$5.359E-10$	$2.393E-10$

Table 14 presents the L_2 and L_∞ norm errors calculated by the proposed LHOC scheme and other reference methods under varying parameters. The results indicate that the LHOC scheme consistently produces the lowest errors, demonstrating its superior performance. Tables 15 and 16 provide the convergence orders of the proposed method in both the spatial and temporal directions. From these tables, it is evident that the method achieves fourth-order accuracy in both directions, in agreement with the theoretical derivation, further validating the effectiveness and consistency of the LHOC scheme.

Table 15. The convergence rate of L_2, L_∞ norm errors of the LHOC scheme for Example 4.5 with $\tau = h$, $T = 2$ at different h and λ.

h	$\lambda = 0.25$			$\lambda = 0.5$		
	L_2	L_∞	$Rate_h$	L_2	L_∞	$Rate_h$
$\frac{1}{4}$	$5.474E-05$	$3.603E-05$	–	$2.404E-01$	$2.801E-01$	–
$\frac{1}{8}$	$3.786E-06$	$2.600E-06$	3.79	$4.156E-02$	$4.949E-02$	2.51
$\frac{1}{16}$	$2.433E-07$	$1.676E-07$	3.95	$4.996E-04$	$6.134E-04$	6.33
$\frac{1}{32}$	$1.540E-08$	$1.061E-08$	3.98	$6.349E-06$	$5.981E-06$	6.68

Table 16. The convergence rate of L_2, L_∞ norm errors of the LHOC scheme for Example 4.5 with $h = \frac{1}{80}$, $T = 2$ at different τ and λ.

τ	$\lambda = 0.25$			$\lambda = 0.5$		
	L_2	L_∞	$Rate_\tau$	L_2	L_∞	$Rate_\tau$
$\frac{1}{5}$	$2.353E-05$	$1.637E-05$	–	$1.557E-01$	$1.660E-01$	–
$\frac{1}{10}$	$1.552E-06$	$1.068E-06$	3.94	$1.538E-02$	$1.831E-02$	3.18
$\frac{1}{20}$	$9.904E-08$	$6.822E-08$	3.97	$5.431E-05$	$5.533E-05$	8.37
$\frac{1}{40}$	$6.260E-09$	$4.308E-09$	3.99	$2.604E-06$	$2.460E-06$	4.49

5. Conclusion

In this study, a finite difference scheme was developed to solve the 1D nonlinear Burgers' equation. Firstly, the fourth-order compact difference scheme and the fourth-order Padé scheme were used to discretize the second-order and first-order spatial derivatives of the equation, respectively. Secondly, for the temporal derivative, a fourth-order backward difference scheme was employed. Lastly, the nonlinear terms in the equation were linearized using a Taylor series expansion, resulting in a linear compact finite difference scheme with fourth-order accuracy in both spatial and temporal directions. The scheme proposed in this paper exhibits higher-order compactness in the spatial direction, which effectively suppresses numerical oscillations and enhances the stability of the numerical solution. Moreover, in the temporal direction, the higher-order accuracy of this scheme makes it more precise than most of the high-order formats reported in the literature. The scheme proposed in this paper is linear and offers significant advantages in terms of computational efficiency. Numerical experiments have verified the effectiveness and accuracy of this method, demonstrating its superior performance. However, there are certain difficulties in analyzing its properties such as stability and convergence. These analytical tasks will become the focus of our subsequent research. The scheme proposed in this paper can be extended to two and three dimensions and provides ideas and methods for dealing with nonlinear problems.

Acknowledgements

This study was conceived and completed by all authors collectively. The authors sincerely appreciate the anonymous reviewers and editors for their valuable suggestions, which significantly improved the quality of this paper.

References

[1] M. Abdullah, M. Yaseen and M. De la Sen, *An efficient collocation method based on Hermite formula and cubic B-splines for numerical solution of the Burgers' equation*, Math. Comput. Simul., 2022, 197, 166–184.

[2] G. Arora and B. Singh, *Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method*, Appl. Math. Comput., 2013, 224, 166–177.

[3] J. M. Burgers, *A mathematical model illustrating the theory of turbulence*, Adv. Appl. Mech., 1948, 1, 171–199.

[4] S. A. Dahy and K. T. Elgindy, *High-order numerical solution of viscous Burgers' equation using an extended Cole-Hopf barycentric Gegenbauer integral pseudospectral method*, Int. J. Comput. Math., 2022, 99(3), 446–464.

[5] K. T. Elgindy, *High-order, stable, and efficient pseudospectral method using barycentric Gegenbauer quadratures*, Appl. Numer. Math., 2017, 113, 1–25.

[6] C. Fan and P. Li, *Generalized finite difference method for solving two-dimensional Burgers' equations*, Procedia Eng., 2014, 79, 55–60.

[7] K. Fujisawa and A. Asada, *Nonlinear parametric sound enhancement through different fluid layer and its application to noninvasive measurement*, Meas., 2016, 94, 726–733.

[8] I. A. Ganaie and V. K. Kukreja, *Numerical solution of Burgers' equation by cubic Hermite collocation method*, Appl. Math. Comput., 2014, 237, 571–581.

[9] F. Gao and C. Chi, *Numerical solution of nonlinear Burgers' equation using high accuracy multiquadric quasi-interpolation*, Appl. Math. Comput., 2014, 229, 414–421.

[10] D. A. Hammada and M. S. El-Azab, *2N order compact finite difference scheme with collocation method for solving the generalized Burger's-Huxley and Burger's-Fisher equations*, Appl. Math. Comput., 2015, 258, 296–311.

[11] M. Hussain, *Hybrid radial basis function methods of lines for the numerical solution of viscous Burgers' equation*, Comput. Appl. Math., 2021, 40(1), 1–49.

[12] S. R. Jena and G. S. Gebremedhin, *Decatic b-spline collocation scheme for approximate solution of Burgers' equation*, Numer. Methods Partial Differ. Equ., 2023, 39, 1851–1869.

[13] Y. D. Jin, J. Zhou, Z. K. Shi, H. L. Zhang and C. P. Wang, *Lattice hydrodynamic model for traffic flow on curved road with passing*, Nonlinear Dyn., 2017, 89, 107–124.

[14] M. Kapoor and V. Joshi, *Numerical solution of coupled 1D Burgers' equation by non-uniform algebraic-hyperbolic B-spline differential quadrature method*, Int. J. Comput. Methods Eng. Sci. Mech., 2022, 24(1), 18–39.

[15] A. Khoshfetrat and M. Abedini, *A hybrid DQ/LMQRBF-DQ approach for numerical solution of Poisson-type and Burger's equations in irregular domain*, Appl. Math. Model., 2012, 36, 1885–1901.

[16] A. Korkmaz and I. Dag, *Cubic B-spline differential quadrature methods and stability for Burgers' equation*, Eng. Comput., 2013, 30, 320–344.

[17] S. Kutluay and Y. Ucar, *Numerical solutions of the coupled Burgers equation by the Galerkin quadratic B-spline finite element method*, Math. Methods Appl. Sci., 2013, 36, 2403–2415.

[18] H. Lai and C. Ma, *A new lattice Boltzmann model for solving the coupled viscous Burgers' equation*, Phys. A., 2014, 395, 445–457.

[19] S. Lele, *Compact finite difference schemes with spectral-like resolution*, J. Comput. Phys., 1992, 103, 16–42.

[20] P. W. Li, *Space-time generalized finite difference nonlinear model for solving unsteady Burgers' equations*, Appl. Math. Lett., 2021, 114, 106896.

[21] M. Maryam, S. Fahimeh, B. Esmail and J. Shahnam, *A local reproducing kernel method accompanied by some different edge improvement techniques: Application to the Burgers' equation*, Iran. J. Sci. Technol. Trans. A Sci., 2018, 42(2), 857–871.

[22] A. K. Mittal, *A space-time pseudospectral method for solving multi-dimensional quasi-linear parabolic partial differential (Burgers') equations*, Appl. Numer. Math., 2024, 195, 39–53.

[23] R. C. Mittal and R. K. Jain, *Numerical solutions of nonlinear Burgers' equation with modified cubic b-splines collocation method*, Appl. Math. Comput., 2012, 218, 7839–7855.

[24] R. C. Mittal and R. Jiwari, *Differential quadrature method for two-dimensional Burgers' equations*, Int. J. Comput. Methods Eng. Sci. Mech., 2009, 10(6), 450–459.

[25] R. C. Mittal and R. Rohila, *A study of one dimensional nonlinear diffusion equations by Bernstein polynomial based differential quadrature method*, J. Math. Chem., 2017, 55(2), 673–695.

[26] R. C. Mittal and A. Tripathi, *Numerical solutions of two-dimensional Burgers' equations using modified B-cubic B-spline finite elements*, Eng. Comput., 2015, 32, 1275–1306.

[27] V. Mukundan and A. Awasthi, *Linearized implicit numerical method for Burgers' equation*, Nonlinear Eng., 2016, 5, 219–234.

[28] M. Palav and V. Pradhan, *Efficient numerical solution of Burgers' equation using collocation method based on re-defined uniform hyperbolic polynomial B-splines*, J. Appl. Math. Comput., 2025, 1865–2085.

[29] K. Pandey, L. Verma and A. K. Verma, *On a finite difference scheme for Burgers' equation*, Appl. Math. Comput., 2009, 215(6), 2206–2214.

[30] K. Shah, *Solution of Burgers' equation in a one-dimensional groundwater recharge by spreading using q-Homotopy analysis method*, Eur. J. Pure Appl. Math., 2016, 9, 114–124.

[31] M. A. Shallal, A. H. Taqi, B. F. Jumaa, H. Rezazadeh and M. Inc, *Numerical solutions to the 1-D Burgers' equation by a cubic Hermite finite element method*, Indian J. Phys., 2022, 96 , 3831–3836.

[32] H. Shukla, M. Tamsir, V. Srivastava and J. Kumar, *Numerical solution of two dimensional coupled viscous Burger equation using modified cubic B-spline differential quadrature method*, AIP Adv., 2014, 117134, 1–10.

[33] V. P. Shyaman, A. Sreelakshmi and A. Awasthi, *A higher order implicit adaptive finite point method for the Burgers' equation*, J. Diff. Eq. Appl., 2023, 29(3), 235–269.

[34] V. Srivastava, S. Singh and M. Awasthi, *Numerical solutions of coupled Burgers' equations by an implicit finite-difference scheme*, AIP Adv., 2013, 082131, 1–7.

[35] C. C. Tsai, Y. T. Shih, Y. T. Lin and H. C. Wang, *Tailored finite point method for solving one-dimensional Burgers' equation*, Int. J. Comput. Math., 2017, 94(4), 800–812.

[36] Y. Uçar, N. M. Yağmurlu and M. K. Yiğit, *Numerical solution of the coupled Burgers equation by trigonometric B-spline collocation method*, Math. Meth. Appl. Sci., 2023, 46(5), 6025–6041.

[37] S. D. Wang, T. F. Ma, L. L. Wu and X. J. Yang, *Two high-order compact finite difference schemes for solving the nonlinear generalized Benjamin-Bona-Mahony-Burgers equation*, Appl. Math. Comput., 2025, 496, 129360.

[38] L. Yang and X. Pu, *Derivation of the Burgers' equation from the gas dynamics*, Commun. Math. Sci., 2016, 14, 671–682.

[39] X. J. Yang, Y. B. Ge and B. Lan, *A class of compact finite difference schemes for solving the 2D and 3D Burgers' equations*, Math. Comput. Simulat., 2021, 185, 510–534.

[40] X. J. Yang, Y. B. Ge and L. Zhang, *A class of high-order compact difference schemes for solving the Burgers' equations*, Appl. Math. Comput., 2019, 358, 394–417.

[41] T. Zhanlav, O. Chuluunbaatar and V. Ulziibayar, *Higher-order accurate numerical solution of unsteady Burgers' equation*, Appl. Math. Comput., 2015, 250, 701–707.

[42] H. Zhu, H. Shu and M. Ding, *Numerical solutions of two-dimensional Burgers' equations by discrete adomian decomposition method*, Comput. Math. Appl., 2010, 60(3), 840–848.

Received November 2024; Accepted April 2025; Available online May 2025.