Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 34(3): 1386–1412.
DOI: 10.3934/era.2026063
Received: 04 November 2025
Revised: 23 January 2026
Accepted: 27 January 2026
Published: 11 February 2026

Research article

A new high-order compact numerical method for the two-dimensional nonlinear Burgers' equation

Shengdi Wang[1], Jiaye Gan[2], Yingnan Qi[1] and Lili Wu[1,*]

[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

* **Correspondence:** Email: wll82003019@nxnu.edu.cn.

Abstract: The Burgers' equation is a key model equation in fluid mechanics and nonlinear physics, widely used to describe the formation and propagation of shock waves. This paper proposes a highly accurate numerical algorithm to solve the two-dimensional Burgers' equation. In the algorithmic framework, the time derivative is discretized using a fourth-order backward finite difference approach. In contrast, the spatial derivatives are processed by a fourth-order Padé scheme and a fourth-order compact finite difference scheme, respectively. To linearize the nonlinear term in the equation, an extrapolation-based linearization technique is introduced, which effectively reduces the computational complexity while maintaining the numerical accuracy. Theoretical analyses of this algorithm demonstrated its unique solvability and stability. Numerous numerical experiments verify the scheme's stability, computational efficiency, and convergence. This approach achieves a fourth-order accuracy in both the temporal and spatial dimensions, thus offering the benefit of maintaining high agreement between the numerical and exact solutions, even when using relatively large time steps, thereby significantly improving the computational efficiency. In addition, the algorithm employs linearization to ease the computational demands, thereby providing a viable and efficient scheme to address high-dimensional nonlinear Burgers' equations.

Keywords: two-dimensional Burgers' equation; fourth-order compact finite difference scheme; fourth-order backward finite difference formula; fourth-order Padé scheme

1. Introduction

The Burgers' equation, an important nonlinear partial differential equation that arises in fluid mechanics, nonlinear acoustics, and gas dynamics, is widely used to describe the formation and propagation mechanisms of shock waves. Relevant applied research includes derivations in gas dynamics [1–5]. Therefore, the development of efficient and stable numerical algorithms to solve this equation has long been a focus of research both domestically and internationally.

For the numerical solution of the one-dimensional Burgers' equation, a wide range of algorithms have been developed to address different computational challenges, including differential quadrature and mesh-free techniques, implicit discretizations and splitting strategies, and compact finite differences. For example, Khoshfetrat and Abedini [6] combined the Differential Quadrature (DQ) method with Local Multiquadric Radial Basis Functions (LMQRBF), thus effectively handling irregular domains via the mesh-free properties of radial basis functions. Mohanty and Schetia [7] introduced an implicit discretization that directly computed first-order spatial derivatives and performed well for singular problems. Seydaoglu et al. [8] employed a high-order splitting algorithm to simplify the treatment of nonlinear terms through operator splitting. Mittal and Rohila [9] introduced a Bernstein polynomial-based differential quadrature method. The cubic Hermite collocation method proposed by Ganaie and Kreja [10] showed excellent robustness at high Reynolds numbers. Yang et al. [11] proposed a linear compact high-order finite difference method, which ensured fourth-order spatial accuracy while maintaining linearity of the computational process. Mukundan and Awasthi [12] designed a linearized semi-discrete method that simplified the solution of nonlinear systems. Zhanlav et al. [13] achieved a rare third-order temporal accuracy, and Miao [14] introduced a high-precision spatiotemporally matched scheme. Moreover, cubic B-spline methods have been widely adopted [15–19]. In terms of the spatiotemporal accuracy, many one-dimensional methods attain fourth-order accuracy in space, whereas their temporal accuracy is often limited to second order.

For two-dimensional Burgers equations, existing numerical methods can be broadly classified into two categories. The first category consists of robust schemes that are widely used in practice, often with relatively a modest spatiotemporal accuracy in practical implementations. Fan and Li [20] proposed a meshless method combined with virtual time integration to avoid complicated mesh generation. Mittal and Jiwari [21] employed the differential quadrature method with fast convergence. Soliman [22] introduced a variational iteration method for non-homogeneous coupled equations to effectively handle nonlinear terms. Zhu [23] used a discrete Adomian decomposition method to obtain series-form approximations. In addition, Zhao et al. [24] developed a local discontinuous Galerkin method with advantages for handling discontinuities. Baharlouei et al. [25] further improved the stability via a hybrid discontinuous Galerkin method. Bahadir [26] presented a fully implicit finite difference scheme known for unconditional stability. More recently, a weak Galerkin finite element method was proposed for the two-dimensional Burgers equation [27], where both semi-discrete and fully discrete schemes were constructed and the existence and uniqueness of the fully discrete solution were proved. Overall, although these methods can effectively handle two-dimensional problems, there remains room for improvement in accurately resolving complex nonlinear wave structures, especially for high Reynolds numbers or in the presence of steep gradients.

The second category focuses on a high spatial accuracy but often suffers from lagging temporal precision. To overcome the limitations of low-order methods, a number of high-resolution spatial discretizations have been developed in recent years. Gottlieb and Wang [28] employed a Fourier collocation spectral method to solve the three-dimensional viscous Burgers equation, and achieved an extremely high spatial spectral accuracy. Yang et al. [29] proposed a high-order compact implicit finite difference scheme while maintaining compactness in multidimensional settings. Peng et al. [30] constructed a fourth-order compact scheme based on nonlinear operators and reduction techniques to enhance the spatial approximation capability. Mittal [31] developed a spatiotemporal pseudospectral method using Lagrange polynomials to improve the global approximation accuracy. In addition, a nonstandard numerical manifold method (NMM) combined with a characteristic Galerkin approach was proposed [32], where the Crank–Nicolson scheme was employed for temporal discretization along the characteristics, and the numerical results indicated good stability and accuracy for high Reynolds numbers. However, a critical review of these methods reveals a common limitation, while the spatial accuracy has been pushed to high orders, the temporal discretization often remains at second order. This phenomenon can be attributed to two primary factors. First, historically, researchers have disproportionately focused on improving the spatial accuracy, often overlooking the numerical degradation caused by spatiotemporal mismatch. Consequently, the potential of high-order spatial operators is frequently compromised by a low-order temporal accuracy. Second, more importantly, for high-dimensional problems, constructing schemes that achieve a high-order accuracy in both space and time presents significant structural challenges. If the spatial and temporal approximations are not effectively coupled, then the construction of the numerical scheme becomes extremely intricate, thus leading to a drastic escalation in algebraic complexity (often referred to as "algebraic explosion"). Due to these difficulties, most existing studies compromise by adopting mature second-order time marching techniques, thus leaving the issue of matched spatiotemporal accuracy unresolved.

To address these challenges, this study proposes a novel linear high-order compact (LHOC) finite difference scheme for the 2D Burgers' equation. The core advantage of the proposed method lies in its ability to achieve a fourth-order temporal accuracy within a fully linear framework, thereby avoiding the heavy computational cost of nonlinear iterations. It is worth noting that recently, several linearized high-order compact schemes were developed for other nonlinear evolution equations without sacrificing the accuracy, such as the Generalized Rosenau-Korteweg-de Vries-Regularized Long Wave equation [33], the Benjamin-Bona-Mahony-Korteweg-de Vries equation [34], and the Generalized Benjamin-Bona-Mahony-Burgers equation [35]. While these works successfully achieved linearity, they typically dealt with the nonlinear term by linearizing the entire term or its coefficients (e.g., $(\frac{u^2}{2})_x$). Distinct from these approaches, our scheme targets the nonlinear convection terms $uu_x + uu_y$ in the 2D Burgers' equation. Instead of linearizing the whole term, we employ an extrapolation strategy specifically for the spatial derivative components u_x and u_y. This allows us to efficiently decouple the nonlinearity while maintaining a high-order precision. Achieving such fourth-order temporal accuracy while preserving a linear solution structure is non-trivial, thus effectively bridging the gap between a high accuracy and the computational efficiency.

Compared with existing methods, the proposed method achieves fourth-order accuracy in both time and space, thereby effectively resolving the issue of a mismatched spatiotemporal accuracy. Additionally, by transforming the nonlinear problem into a linear system, the method reduces the computational complexity, allows for larger time steps, and significantly improves the computational efficiency. Both a theoretical analysis and numerical experiments demonstrate that this scheme possesses conditional unique solvability and stability.

The remainder of this paper is organized as follows: Section 2 presents the model problem, Section 3 constructs the numerical scheme, Section 4 provides the theoretical analysis, Section 5 reports numerical experiments, and Section 6 concludes the paper.

2. Model equation

In this paper, the finite difference method is applied to solve the following two-dimensional Burgers' equation:

$$u_t + uu_x + uu_y = \varepsilon(u_{xx} + u_{yy}), \qquad (x, y, t) \in D_1 \times [T_0, T]. \tag{2.1}$$

Here, ε is defined as the reciprocal of the Reynolds number (Re), which is always positive.

In addition, Eq (2.1) has the initial conditions

$$u(x, y, T_0) = \eta_1(x, y), \qquad (x, y) \in D_1, \tag{2.2}$$

and boundary conditions

$$u(x, y, t) = \psi_1(x, y, t), \qquad (x, y, t) \in \partial D_1 \times [T_0, T]. \tag{2.3}$$

Here, ∂D_1 denotes the boundary of D_1.

3. Linear high-order compact finite difference scheme

Set $D_1 = \{(x, y) : a \leq x, y \leq b\}$, where a and b are constants. Consider the domain $[a, b] \times [a, b] \times [T_0, T]$, which is discretized into N_x and N_y intervals in the x and y-directions and M intervals in the time direction, respectively, with spatial step lengths $h_x = (b - a)/N_x$ and $h_y = (b - a)/N_y$. We set $h_x = h_y = h$. The temporal step is chosen as $\tau = (T - T_0)/M$. Let $u(x, y, t)$ be the exact solution, $U^n_{i,j}$ represent the numerical approximation of $u(x, y, t)$ at the grid point (x_i, y_j, t_n), and $Z_h = \{u_{i,j}, i = 0, 1, 2, \cdots, N_x, j = 0, 1, 2, \cdots, N_y\}$. The notation for mesh functions $u, v \in Z_h$ is introduced as follows:

$$\delta_x u^n_{i,j} = \frac{u^n_{i+1,j} - u^n_{i-1,j}}{2h_x}, \quad \delta_y u^n_{i,j} = \frac{u^n_{i,j+1} - u^n_{i,j-1}}{2h_y}, \quad \delta^+_t u^n_{i,j} = \frac{u^{n+1}_{i,j} - u^n_{i,j}}{\tau}$$
$$\delta^2_x u^n_{i,j} = \frac{u^n_{i+1,j} - 2u^n_{i,j} + u^n_{i-1,j}}{h^2_x}, \quad \delta^2_y u^n_{i,j} = \frac{u^n_{i,j+1} - 2u^n_{i,j} + u^n_{i,j-1}}{h^2_y}.$$

The fourth-order compact finite difference operators $\mathcal{A}$ and $\mathcal{B}$ are defined as follows:

$$\begin{aligned}
\mathcal{A}_x u^n_{i,j} &= \frac{1}{12}(u^n_{i,j+1} + 10u^n_{i,j} + u^n_{i,j-1}) = (1 + \frac{h^2_x}{12}\delta^2_x)u^n_{i,j},\\
\mathcal{A}_y u^n_{i,j} &= \frac{1}{12}(u^n_{i,j+1} + 10u^n_{i,j} + u^n_{i,j-1}) = (1 + \frac{h^2_y}{12}\delta^2_y)u^n_{i,j},\\
\mathcal{B}_x u^n_{i,j} &= \frac{1}{6}(u^n_{i+1,j} + 4u^n_{i,j} + u^n_{i-1,j}) = (1 + \frac{h^2_x}{6}\delta^2_x)u^n_{i,j},\\
\mathcal{B}_y u^n_{i,j} &= \frac{1}{6}(u^n_{i,j+1} + 4u^n_{i,j} + u^n_{i,j-1}) = (1 + \frac{h^2_y}{6}\delta^2_y)u^n_{i,j},
\end{aligned}$$

which satisfy

$$\mathcal{A}_x(u_{xx})^n_{i,j} = \delta^2_x u^n_{i,j} + O(h^4_x), \qquad \mathcal{A}_y(u_{yy})^n_{i,j} = \delta^2_y u^n_{i,j} + O(h^4_y),$$

$$\mathcal{B}_x(u_x)^n_{i,j} = \delta_x u^n_{i,j} + O(h_x^4), \qquad \mathcal{B}_y(u_y)^n_{i,j} = \delta_y u^n_{i,j} + O(h_y^4). \tag{3.1}$$

The above is the fourth-order Padé scheme [36] to solve the first and second derivatives.

From the above definition, we can see that $\mathcal{A}_x$ and $\mathcal{B}_x$ are defined along the x-axis, while $\mathcal{A}_y$ and $\mathcal{B}_y$ are defined along the y-axis. It follows that $\mathcal{A}$ and $\mathcal{B}$ are diagonally dominant, and they have invertible operators $\mathcal{A}^{-1}$ and $\mathcal{B}^{-1}$, which satisfy the following:

$$\begin{aligned}
(u_{xx})^n_{i,j} &= \mathcal{A}_x^{-1}\delta_x^2 u^n_{i,j} + O(h_x^4), \qquad (u_{yy})^n_{i,j} = \mathcal{A}_y^{-1}\delta_y^2 u^n_{i,j} + O(h_y^4),\\
(u_x)^n_{i,j} &= \mathcal{B}_x^{-1}\delta_x u^n_{i,j} + O(h_x^4), \qquad (u_y)^n_{i,j} = \mathcal{B}_y^{-1}\delta_y u^n_{i,j} + O(h_y^4).
\end{aligned} \tag{3.2}$$

The above is the fourth-order compact finite difference formula to solve the first and second derivatives. Next, we construct a numerical scheme to solve Eqs (2.1)–(2.3). The value of Equation (2.1) at (x_i, y_j, t_{n+1}) is as follows:

$$(u_t)^{n+1}_{i,j} + u^{n+1}_{i,j}(u_x)^{n+1}_{i,j} + u^{n+1}_{i,j}(u_y)^{n+1}_{i,j} = \varepsilon[(u_{xx})^{n+1}_{i,j} + (u_{yy})^{n+1}_{i,j}]. \tag{3.3}$$

For the first-order derivative with respect to time in Eq (3.3), we adopt the following five-level fourth-order backward difference formula (BDF-4) for discretization:

$$\Delta_t u^n_{i,j} = \frac{25u^n_{i,j} - 48u^{n-1}_{i,j} + 36u^{n-2}_{i,j} - 16u^{n-3}_{i,j} + 3u^{n-4}_{i,j}}{12\tau} + O(\tau^4). \tag{3.4}$$

Substituting Eq (3.4) into (3.3), we have the following:

$$\Delta_t u^{n+1}_{i,j} + \left[(u_x)^{n+1}_{i,j} + (u_y)^{n+1}_{i,j}\right]u^{n+1}_{i,j} = \varepsilon\left[(u_{xx})^{n+1}_{i,j} + (u_{yy})^{n+1}_{i,j}\right] + O(\tau^4). \tag{3.5}$$

Next, consider the discretization of spatial derivatives on a uniform grid. In Eq (3.6), the second-order spatial derivative is approximated through a fourth-order compact finite difference approach, the following:

$$\Delta_t u^{n+1}_{i,j} + \left[(u_x)^{n+1}_{i,j} + (u_y)^{n+1}_{i,j}\right]u^{n+1}_{i,j} = \varepsilon\left(\mathcal{A}_x^{-1}\delta_x^2 u^{n+1}_{i,j} + \mathcal{A}_y^{-1}\delta_y^2 u^{n+1}_{i,j}\right) + O(\tau^4 + h^4). \tag{3.6}$$

Multiplying both sides of the above equation by $\mathcal{A}_x\mathcal{A}_y$ gives the following:

$$\mathcal{A}_x\mathcal{A}_y\Delta_t u^{n+1}_{i,j} + \mathcal{A}_x\mathcal{A}_y\left[(u_x)^{n+1}_{i,j} + (u_y)^{n+1}_{i,j}\right]u^{n+1}_{i,j} = \varepsilon\left[\mathcal{A}_y\delta_x^2 u^{n+1}_{i,j} + \mathcal{A}_x\delta_y^2 u^{n+1}_{i,j}\right] + O(\tau^4 + h^4). \tag{3.7}$$

Substituting the definitions of the central difference operator and the fourth-order compact difference operator into Eq (3.7), we have the following:

$$\begin{aligned}
&\left[\frac{625}{36} + \frac{25\tau}{3}(U_x)^{n+1}_{i,j} + \frac{25\tau}{3}(U_y)^{n+1}_{i,j} + \frac{40\tau\varepsilon}{h^2}\right]U^{n+1}_{i,j} + \left[\frac{125}{72} + \frac{5\tau}{6}(U_x)^{n+1}_{i+1,j} + \frac{5\tau}{6}(U_y)^{n+1}_{i+1,j} - \frac{8\tau\varepsilon}{h^2}\right]U^{n+1}_{i+1,j}\\
&+\left[\frac{125}{72} + \frac{5\tau}{6}(U_x)^{n+1}_{i-1,j} + \frac{5\tau}{6}(U_y)^{n+1}_{i-1,j} - \frac{8\tau\varepsilon}{h^2}\right]U^{n+1}_{i-1,j} + \left[\frac{125}{72} + \frac{5\tau}{6}(U_x)^{n+1}_{i,j+1} + \frac{5\tau}{6}(U_y)^{n+1}_{i,j+1} - \frac{8\tau\varepsilon}{h^2}\right]U^{n+1}_{i,j+1}\\
&+\left[\frac{125}{72} + \frac{5\tau}{6}(U_x)^{n+1}_{i,j-1} + \frac{5\tau}{6}(U_y)^{n+1}_{i,j-1} - \frac{8\tau\varepsilon}{h^2}\right]U^{n+1}_{i,j-1} + \left[\frac{25}{144} + \frac{\tau}{12}(U_x)^{n+1}_{i+1,j+1} + \frac{\tau}{12}(U_y)^{n+1}_{i+1,j+1} - \frac{2\tau\varepsilon}{h^2}\right]U^{n+1}_{i+1,j+1}\\
&\qquad+\left[\frac{25}{144} + \frac{\tau}{12}(U_x)^{n+1}_{i-1,j+1} + \frac{\tau}{12}(U_y)^{n+1}_{i-1,j+1} - \frac{2\tau\varepsilon}{h^2}\right]U^{n+1}_{i-1,j+1}
\end{aligned}$$

$$+\left[\frac{25}{144}+\frac{\tau}{12}(U_x)_{i+1,j-1}^{n+1}+\frac{\tau}{12}(U_y)_{i+1,j-1}^{n+1}-\frac{2\tau\varepsilon}{h^2}\right]U_{i+1,j-1}^{n+1}$$

$$+\left[\frac{25}{144}+\frac{\tau}{12}(U_x)_{i-1,j-1}^{n+1}+\frac{\tau}{12}(U_y)_{i-1,j-1}^{n+1}-\frac{2\tau\varepsilon}{h^2}\right]U_{i-1,j-1}^{n+1}$$

$$=(48U_{i,j}^{n}-36U_{i,j}^{n-1}+16U_{i,j}^{n-2}-3U_{i,j}^{n-3})+\frac{h^2}{12}\delta_x^2(48U_{i,j}^{n}-36U_{i,j}^{n-1}+16U_{i,j}^{n-2}-3U_{i,j}^{n-3})+\frac{h^4}{144}\delta_x^2\delta_y^2$$

$$(48U_{i,j}^{n}-36U_{i,j}^{n-1}+16U_{i,j}^{n-2}-3U_{i,j}^{n-3})+\frac{h^2}{12}\delta_y^2(48U_{i,j}^{n}-36U_{i,j}^{n-1}+16U_{i,j}^{n-2}-3U_{i,j}^{n-3}). \tag{3.8}$$

Equation (3.8) is a nonlinear numerical method to solve Eqs (2.1)–(2.3). To obtain a linearized solution scheme, we will apply an appropriate linearization strategy to linearize Eq (3.8) below.

Next, we apply the Taylor series expansion method to perform a linearization approximation of $(u_x)^{n+1}$. Let $w^{n+1}=(u_x)^{n+1}$, then, the following expansions are obtained:

$$\begin{cases} w^{n+1}=w^n+\tau(w_t)^n+\frac{\tau^2}{2}(w_{tt})^n+\frac{\tau^3}{6}(w_{ttt})^n+O(\tau^4) & ① \\ w^{n-1}=w^n-\tau(w_t)^n+\frac{\tau^2}{2}(w_{tt})^n-\frac{\tau^3}{6}(w_{ttt})^n+O(\tau^4) & ② \\ w^{n-2}=w^n-2\tau(w_t)^n+\frac{(2\tau)^2}{2}(w_{tt})^n-\frac{(2\tau)^3}{6}(w_{ttt})^n+O(\tau^4) & ③ \\ w^{n-3}=w^n-3\tau(w_t)^n+\frac{(3\tau)^2}{2}(w_{tt})^n-\frac{(3\tau)^3}{6}(w_{ttt})^n+O(\tau^4). & ④ \end{cases}$$

The final expression is derived via the linear combination ①+a②+b③+c④ as follows:

$$\begin{aligned} &w^{n+1}+aw^{n-1}+bw^{n-2}+cw^{n-3} \\ &=(1+a+b+c)w^n+\tau(1-a-2b-3c)(w_t)^n+\tfrac{\tau^2}{2}(1+a+4b+9c)(w_{tt})^n \\ &\quad+\tfrac{\tau^3}{6}(1-a-8b-27c)(w_{ttt})^n+O(\tau^4). \qquad ⑤ \end{aligned}$$

To ensure the desired accuracy, the coefficients are determined by solving the system

$$\begin{cases} 1+a+4b+9c=0 \\ 1-a-2b-3c=0 \\ 1-a-8b-27c=0 \\ 1+a+b+c=4, \end{cases}$$

which yields the solution $a=6, b=-4$, and $c=1$. Substitute the values of parameters a, b, and c into ⑤, and simultaneously substitute the definition of w^{n+1} to obtain the following:

$$(u_x)_{i,j}^{n+1}=4(u_x)_{i,j}^{n}-6(u_x)_{i,j}^{n-1}+4(u_x)_{i,j}^{n-2}-(u_x)_{i,j}^{n-3}+O(\tau^4). \tag{3.9}$$

The same can be obtained as follows:

$$(u_y)_{i,j}^{n+1}=4(u_y)_{i,j}^{n}-6(u_y)_{i,j}^{n-1}+4(u_y)_{i,j}^{n-2}-(u_y)_{i,j}^{n-3}+O(\tau^4). \tag{3.10}$$

By inserting Eqs (3.9) and (3.10) into (3.8), we obtain a linear scheme to solve Eqs (2.1)–(2.3). Then the first derivatives in space from Eqs (3.9) and (3.10) are approximated using the fourth-order Padé scheme. At the boundary points, the first-order derivatives in the x and y directions can be approximated by adopting the boundary schemes with fourth-order accuracy as follows [35]:

$$
\begin{aligned}
&\frac{2}{3}(U_x)_{0,j}+\frac{1}{3}(U_x)_{1,j}=\\
&\frac{1}{h_x}\left(-\frac{143}{90}U_{0,j}+\frac{107}{36}U_{1,j}-\frac{8}{3}U_{2,j}+\frac{17}{9}U_{3,j}-\frac{13}{18}U_{4,j}+\frac{7}{60}U_{5,j}\right),\\
&\frac{1}{3}(U_x)_{N_x-1,j}+\frac{2}{3}(U_x)_{N_x,j}=\\
&\frac{1}{h_x}\left(\frac{143}{90}U_{N_x,j}-\frac{107}{36}U_{N_x-1,j}+\frac{8}{3}U_{N_x-2,j}-\frac{17}{9}U_{N_x-3,j}+\frac{13}{18}U_{N_x-4,j}-\frac{7}{60}U_{N_x-5,j}\right),\\
&\frac{2}{3}(U_y)_{i,0}+\frac{1}{3}(U_y)_{i,1}=\\
&\frac{1}{h_y}\left(-\frac{143}{90}U_{i,0}+\frac{107}{36}U_{i,1}-\frac{8}{3}U_{i,2}+\frac{17}{9}U_{i,3}-\frac{13}{18}U_{i,4}+\frac{7}{60}U_{i,5}\right),\\
&\frac{1}{3}(U_y)_{i,N_y-1}+\frac{2}{3}(U_y)_{i,N_y}=\\
&\frac{1}{h_y}\left(\frac{143}{90}U_{i,N_y}-\frac{107}{36}U_{i,N_y-1}+\frac{8}{3}U_{i,N_y-2}-\frac{17}{9}U_{i,N_y-3}+\frac{13}{18}U_{i,N_y-4}-\frac{7}{60}U_{i,N_y-5}\right),
\end{aligned}
\tag{3.11}
$$

where $i=0,1,2,\cdots,N_x$, $j=0,1,2,\cdots,N_y$.

Therefore, this study introduces a linear high-order compact finite difference scheme (LHOC) to solve Eqs (2.1)–(2.3), which achieves a fourth-order accuracy in both the temporal and spatial directions. Since the scheme has a five-layer structure, a startup algorithm is required to generate the numerical solutions from the first to the third layers. This can be achieved by using a two-layer nonlinear scheme, for the computation of u^1, u^2, u^3.

Equation (2.1), when assessed at the intermediate time level $n+\frac{1}{2}$ with $g=\frac{1}{2}u^2$, gives the following:

$$
(u_t)_{i,j}^{n+\frac{1}{2}}+(g_x)_{i,j}^{n+\frac{1}{2}}+(g_y)_{i,j}^{n+\frac{1}{2}}=\varepsilon\left[(u_{xx})_{i,j}^{n+\frac{1}{2}}+(u_{yy})_{i,j}^{n+\frac{1}{2}}\right].
\tag{3.12}
$$

The derivative in the time direction in the above equation can be discretized using the Crank-Nicolson scheme, obtaining the following:

$$
\begin{aligned}
&\delta_t^+u_{i,j}^n+\frac{1}{2}\left[(g_x)_{i,j}^{n+1}+(g_x)_{i,j}^n+(g_y)_{i,j}^{n+1}+(g_y)_{i,j}^n\right]\\
&=\frac{\varepsilon}{2}\left[(u_{xx})_{i,j}^{n+1}+(u_{xx})_{i,j}^n+(u_{yy})_{i,j}^{n+1}+(u_{yy})_{i,j}^n\right]+O(\tau^2).
\end{aligned}
\tag{3.13}
$$

In Eq (3.13), the second-order derivative of the function is approximated using a fourth-order compact difference scheme, thus, we obtain the following:

$$
\begin{aligned}
&\mathcal{A}_x\mathcal{A}_y\delta_t^+u_{i,j}^n+\frac{1}{2}\mathcal{A}_x\mathcal{A}_y\left[(g_x)_{i,j}^{n+1}+(g_x)_{i,j}^n+(g_y)_{i,j}^{n+1}+(g_y)_{i,j}^n\right]\\
&=\frac{\varepsilon}{2}\left[\mathcal{A}_y\delta_x^2u_{i,j}^{n+1}+\mathcal{A}_y\delta_x^2u_{i,j}^n+\mathcal{A}_x\delta_y^2u_{i,j}^{n+1}+\mathcal{A}_x\delta_y^2u_{i,j}^n\right]+O(\tau^2+h^4).
\end{aligned}
\tag{3.14}
$$

Substituting the definitions of the fourth-order compact difference operators $\mathcal{A}_x$ and $\mathcal{A}_y$ yields the following:

$$\begin{aligned}
&\frac{1}{\tau}(1+\frac{h^2}{12}\delta_x^2)(1+\frac{h^2}{12}\delta_y^2)u_{i,j}^{n+1}-\frac{\varepsilon}{2}(1+\frac{h^2}{12}\delta_y^2)\delta_x^2u_{i,j}^{n+1}-\frac{\varepsilon}{2}(1+\frac{h^2}{12}\delta_x^2)\delta_y^2u_{i,j}^{n+1}\\
&=\frac{1}{\tau}(1+\frac{h^2}{12}\delta_x^2)(1+\frac{h^2}{12}\delta_y^2)u_{i,j}^{n}+\frac{\varepsilon}{2}(1+\frac{h^2}{12}\delta_y^2)\delta_x^2u_{i,j}^{n}+\frac{\varepsilon}{2}(1+\frac{h^2}{12}\delta_x^2)\delta_y^2u_{i,j}^{n}\\
&-\frac{1}{2}(1+\frac{h^2}{12}\delta_x^2)(1+\frac{h^2}{12}\delta_y^2)\left[(g_x)_{i,j}^{n+1}+(g_x)_{i,j}^{n}+(g_y)_{i,j}^{n+1}+(g_y)_{i,j}^{n}\right]+O(\tau^2+h^4).
\end{aligned}\tag{3.15}$$

Then, by replacing the above equation with the central difference operator's definition, we can obtain the following:

$$\begin{aligned}
&(\frac{25}{36\tau}-\frac{5\varepsilon}{3h^2})U_{i,j}^{n+1}+(\frac{5}{72\tau}-\frac{\varepsilon}{3h^2})(U_{i+1,j}^{n+1}+U_{i-1,j}^{n+1}+U_{i,j+1}^{n+1}+U_{i,j-1}^{n+1})\\
&+(\frac{1}{144\tau}-\frac{\varepsilon}{12h^2})(U_{i+1,j+1}^{n+1}+U_{i-1,j+1}^{n+1}+U_{i+1,j-1}^{n+1}+U_{i-1,j-1}^{n+1})=\\
&(\frac{25}{36\tau}+\frac{5\varepsilon}{3h^2})U_{i,j}^{n}+(\frac{5}{72\tau}+\frac{\varepsilon}{3h^2})(U_{i+1,j}^{n}+U_{i-1,j}^{n}+U_{i,j+1}^{n}+U_{i,j-1}^{n})\\
&+(\frac{1}{144\tau}+\frac{\varepsilon}{12h^2})(U_{i+1,j+1}^{n}+U_{i-1,j+1}^{n}+U_{i+1,j-1}^{n}+U_{i-1,j-1}^{n+1})\\
&-\frac{1}{2}(1+\frac{h^2}{12}\delta_x^2)(1+\frac{h^2}{12}\delta_y^2)\left[(G_x)_{i,j}^{n+1}+(G_x)_{i,j}^{n}+(G_y)_{i,j}^{n+1}+(G_y)_{i,j}^{n}\right],
\end{aligned}\tag{3.16}$$

Here, $G=\frac{1}{2}U^2$, $(G_x)_{i,j}$, and $(G_y)_{i,j}$ in the above equation are computed using Eqs (3.2) and (3.11).

Equation (3.16) constitutes a two-level nonlinear fourth-order finite difference scheme. It attains second-order and fourth-order accuracies in time and space, respectively. To ensure that the temporal accuracy of the startup scheme (19) is consistent with that of the LHOC scheme during the computation of the first three layers u^1, u^2, u^3, the Richardson extrapolation is applied to enhance its temporal accuracy to fourth-order as follows:

$$\hat{u}_{i,j}^n(\tau)=\frac{1}{3}\left[4u_{i,j}^{2n}(\frac{\tau}{2})-u_{i,j}^n(\tau)\right],\quad 0\leqslant i\leqslant N_x,\quad 0\leqslant j\leqslant N_y,\quad 1\leqslant n\leqslant 3.\tag{3.17}$$

In Eq (3.17), $\hat{u}_{i,j}^n(\tau)$ denotes the solution at the n-th time level following the Richardson extrapolation. Here, $u_{i,j}^n(\tau)$ and $u_{i,j}^{2n}(\frac{\tau}{2})$ represent the solutions obtained at the n-th time level through the startup scheme (19) with time steps τ and $\frac{\tau}{2}$, respectively.

The complete computational framework to solve the 2D Burgers' equation using the proposed scheme is outlined in Algorithm 1.

4. Analysis of unique solvability and linear stability

In this section, we conduct a theoretical analysis on the unique solvability and linear stability of the proposed LHOC scheme.

4.1. Unique solvability analysis

By assembling the discretized Eq (4.1) for all (i, j) and incorporating boundary conditions, we obtain a linear system. The unknown nodal values are arranged into a vector $\mathbf{U}^{n+1}=[(\mathbf{U}_1^{n+1})^T,\ldots,(\mathbf{U}_{N_y-1}^{n+1})^T]^T$,

Algorithm 1 Linear high-order compact (LHOC) scheme for 2D Burgers' equation

```
Require: Initial condition (Eq (2.2)), Dirichlet boundary conditions (Eq (2.3)), parameters $\tau, h, \varepsilon, T$.
Ensure: Numerical solution set $\{U^n_{i,j}\}$.
Step 1: Initialization
Set $n = 0$. Compute initial values $U^0_{i,j}$ using Eq (2.2) for $i = 0, \dots, N_x$, $j = 0, \dots, N_y$.
Step 2: Startup phase (Generate $U^1, U^2, U^3$)
for $k = 1$ to 3 do
    Solve Eq (3.16) with $\Delta t = \tau$ to obtain $U_{(\tau)}$.
    Solve Eq (3.16) with $\Delta t = \frac{\tau}{2}$ for two consecutive sub-steps to obtain $U_{(\tau/2)}$.
        ▷ Note: Spatial derivatives via Padé schemes (5) and (14) (Thomas algorithm); implicit system
        via Gauss-Seidel.
    Compute final $U^k$ using Richardson extrapolation formula (Eq (2.17)).
    Apply Dirichlet boundary conditions (Eq (2.3)).
end for
Step 3: Main linear time-marching (for $n = 4$ to int$(T/\tau)$)
for $n = 4$ to int$(T/\tau)$ do
    Compute derivatives $U_x, U_y$ for levels $n-1, \dots, n-4$ using Padé schemes (5) and (14) via
    Thomas algorithm.
    Calculate extrapolated coefficients for current level $n$ using Eqs (3.9) and (3.10):
        $(U_x)^n = 4(U_x)^{n-1} - 6(U_x)^{n-2} + 4(U_x)^{n-3} - (U_x)^{n-4}$
        $(U_y)^n = 4(U_y)^{n-1} - 6(U_y)^{n-2} + 4(U_y)^{n-3} - (U_y)^{n-4}$
    Solve the discretized system (Eq 2.8) using Gauss-Seidel linear iteration to obtain $U^n$.
    Loop until convergence: $\left|U^{n(s+1)}_{i,j} - U^{n(s)}_{i,j}\right| < 10^{-14}$.
    Calculate boundary values for $U^n$ using Dirichlet condition Eq (2.3).
end for
Step 4: Output
Return the final solution set $\{U^n_{i,j}\}_{n=0}^{\text{int}(T/\tau)}$.
```

thus, we have the following:

$$\mathbf{A}\mathbf{U}^{n+1} = \mathbf{F}, \tag{4.1}$$

where $\mathbf{A}$ is a block tridiagonal matrix of size $(N_x - 1)(N_y - 1) \times (N_x - 1)(N_y - 1)$

$$A = \begin{pmatrix} D_1 & U_1 & & & \\ L_2 & D_2 & U_2 & & \\ & \ddots & \ddots & \ddots & \\ & & L_{N_y-2} & D_{N_y-2} & U_{N_y-2} \\ & & & L_{N_y-1} & D_{N_y-1} \end{pmatrix}, \tag{4.2}$$

where each sub-matrix D_j, L_j, U_j is of size $(N_x - 1) \times (N_x - 1)$. The coefficients from the 9-point stencil in Eq (4.1) map to these blocks as follows: $D_j = \text{tridiag}(w_{i,j},\ d_{i,j},\ e_{i,j})$, $U_j = \text{tridiag}(nw_{i,j},\ n_{i,j},\ ne_{i,j})$, and $L_j = \text{tridiag}(sw_{i,j},\ s_{i,j},\ se_{i,j})$. The coefficients are defined as follows:

$$d_{i,j} = \frac{625}{36} + \frac{25\tau}{3}(f_{i,j} + g_{i,j}) + \frac{40\tau\varepsilon}{h^2},$$

$$
\begin{aligned}
e_{i,j} &= \frac{125}{72} + \frac{5\tau}{6}(f_{i+1,j} + g_{i+1,j}) - \frac{8\tau\varepsilon}{h^2}, & w_{i,j} &= \frac{125}{72} + \frac{5\tau}{6}(f_{i-1,j} + g_{i-1,j}) - \frac{8\tau\varepsilon}{h^2},\\
n_{i,j} &= \frac{125}{72} + \frac{5\tau}{6}(f_{i,j+1} + g_{i,j+1}) - \frac{8\tau\varepsilon}{h^2}, & s_{i,j} &= \frac{125}{72} + \frac{5\tau}{6}(f_{i,j-1} + g_{i,j-1}) - \frac{8\tau\varepsilon}{h^2},\\
ne_{i,j} &= \frac{25}{144} + \frac{\tau}{12}(f_{i+1,j+1} + g_{i+1,j+1}) - \frac{2\tau\varepsilon}{h^2}, & nw_{i,j} &= \frac{25}{144} + \frac{\tau}{12}(f_{i-1,j+1} + g_{i-1,j+1}) - \frac{2\tau\varepsilon}{h^2},\\
se_{i,j} &= \frac{25}{144} + \frac{\tau}{12}(f_{i+1,j-1} + g_{i+1,j-1}) - \frac{2\tau\varepsilon}{h^2}, & sw_{i,j} &= \frac{25}{144} + \frac{\tau}{12}(f_{i-1,j-1} + g_{i-1,j-1}) - \frac{2\tau\varepsilon}{h^2},
\end{aligned}
$$

where $f_{i,j} = (U_x)_{i,j}^{n+1}$ and $g_{i,j} = (U_y)_{i,j}^{n+1}$.

It is important to note that although the linear system implies a 9-point stencil that involves diagonal neighbors, the scheme is strictly confined within the immediate one-step neighborhood $(x_{i\pm1}, y_{j\pm1})$. This distinguishes it from standard explicit fourth-order schemes, which typically require a wider 5×5 stencil $(x_{i\pm2}, y_{j\pm2})$. In the proposed method, the diagonal nodes are utilized to approximate the high-order cross-derivative terms, which allows a fourth-order accuracy to be achieved within the minimal 3×3 localized domain. Consequently, this structure is classified as a compact scheme.

Theorem 1 (Conditional existence and uniqueness)**.** The linear system $\mathbf{A}\mathbf{U}^{n+1} = \mathbf{F}$ is conditionally uniquely solvable. For given spatial step sizes and bounded gradients $M = \max_{i,j}(|f_{i,j}| + |g_{i,j}|)$, there exists a time step $\tau > 0$ such that $\mathbf{A}$ is strictly diagonally dominant.

Proof. First, we have the following:

$$|d_{i,j}| \geqslant \left(\frac{625}{36} + \frac{40\tau\varepsilon}{h^2}\right) - \frac{25\tau}{3}M. \tag{4.3}$$

$$\sum_{(p,q)\in\{(i\pm1,j),(i,j\pm1)\}} |a_{p,q}| \leqslant 4\left(\frac{125}{72} + \frac{5\tau}{6}M + \frac{8\tau\varepsilon}{h^2}\right) = \frac{500}{72} + \frac{10\tau}{3}M + \frac{32\tau\varepsilon}{h^2}. \tag{4.4}$$

$$\sum_{(p,q)\in\{(i\pm1,j\pm1)\}} |a_{p,q}| \leqslant 4\left(\frac{25}{144} + \frac{\tau}{12}M + \frac{2\tau\varepsilon}{h^2}\right) = \frac{100}{144} + \frac{\tau}{3}M + \frac{8\tau\varepsilon}{h^2}. \tag{4.5}$$

Thus,

$$\sum_{q\neq p} |a_{pq}| \leqslant \frac{275}{36} + \frac{11\tau}{3}M + \frac{40\tau\varepsilon}{h^2}. \tag{4.6}$$

The strict diagonal dominance condition $|d_{i,j}| > \sum |a_{pq}|$ leads to the following:

$$\left(\frac{625}{36} + \frac{40\tau\varepsilon}{h^2}\right) - \frac{25\tau}{3}M > \frac{275}{36} + \frac{11\tau}{3}M + \frac{40\tau\varepsilon}{h^2}. \tag{4.7}$$

The inequality simplifies to the following:

$$
\begin{aligned}
\frac{350}{36} &> 12\tau M,\\
\tau &< \frac{175}{216M}.
\end{aligned} \tag{4.8}
$$

Since M is bounded, a valid τ always exists, thus ensuring that $\mathbf{A}$ is non-singular and the solution uniquely exists. □

4.2. Linear stability analysis

In this section, the linear stability of the proposed LHOC scheme is investigated. We utilize the Von Neumann analysis method combined with the frozen coefficient technique.

Lemma 1. (See [37]) The stability region S for the BDF-4 formula possesses the following properties:

(i) The stability region contains the entire negative real axis $(-\infty, 0]$.
(ii) The method is zero-stable but not A-stable. The stability region is bounded in the right half-plane, particularly along the positive real axis. Therefore, for a real eigenvalue $\lambda > 0$, the time step τ must satisfy strict stability constraints.

Theorem 2 (Conditional stability)**.** The LHOC difference scheme (3.8) for the 2D Burgers' equation is conditionally stable.

Proof. Let $Q_1(x, y) = (U_x)_{i,j}^{n+1}$ and $Q_2(x, y) = (U_y)_{i,j}^{n+1}$; then,

$$\begin{aligned} Q_1 &= 4(U_x)^n - 6(U_x)^{n-1} + 4(U_x)^{n-2} - (U_x)^{n-3}, \\ Q_2 &= 4(U_y)^n - 6(U_y)^{n-1} + 4(U_y)^{n-2} - (U_y)^{n-3}. \end{aligned} \tag{4.9}$$

The nonlinear problem at time level $n + 1$ is transformed into a linear equation with variable coefficients as follows:

$$U_t + Q_1 U + Q_2 U = \varepsilon(U_{xx} + U_{yy}). \tag{4.10}$$

Let $e_{i,j}^n$ denote the error of the numerical solution at the grid point (x_i, y_j) at time level n. We assume that the error function can be expressed as a Fourier series summation over the computational domain as follows:

$$e_{i,j}^n = \sum_{k_x} \sum_{k_y} \hat{e}_{k_x,k_y}^n \exp\left(\mathbf{i}(\theta_x i + \theta_y j)\right), \tag{4.11}$$

where $\mathbf{i} = \sqrt{-1}$ is the imaginary unit, $\hat{e}_{k_x,k_y}^n$ is the amplitude of the Fourier component, and $\theta_x = k_x h_x, \theta_y = k_y h_y$ are the phase angles with wavenumbers k_x, k_y.

Since Eq (4.9) is linear with constant coefficients, the Fourier modes decouple. Therefore, it suffices to analyze the stability behavior of a single typical Fourier component as follows:

$$e_{i,j}^n = \xi^n \exp\left(\mathbf{i}(\theta_x i + \theta_y j)\right), \tag{4.12}$$

where ξ is the amplification factor of the mode.

First, for the reaction operator λ_1, we have the following:

$$\lambda_1 = Q_1 + Q_2. \tag{4.13}$$

Second, for the diffusion operator λ_2, since $u_{xx} = \mathcal{A}_x^{-1}\delta_x^2 u_{i,j}$ and $u_{yy} = \mathcal{A}_y^{-1}\delta_y^2 u_{i,j}$, we have the following:

$$\lambda_2 = \varepsilon \left[\frac{12(\cos\theta_x - 1)}{h_x^2(5 + \cos\theta_x)} + \frac{12(\cos\theta_y - 1)}{h_y^2(5 + \cos\theta_y)} \right]. \tag{4.14}$$

Since $\cos\theta \leqslant 1$ and $\varepsilon > 0$, it holds that $\lambda_2 \leqslant 0$ for all θ.

Substituting these spectral symbols into the BDF-4 time discretization scheme yields the following characteristic equation:

$$\frac{1}{12\tau}(25\xi^4 - 48\xi^3 + 36\xi^2 - 16\xi + 3) + \lambda_1\xi^4 = \lambda_2\xi^4, \tag{4.15}$$

Rearranging the equation for the generalized eigenvalue z yields the followings:

$$[25 - 12\tau(\lambda_2 - \lambda_1)]\xi^4 - 48\xi^3 + 36\xi^2 - 16\xi + 3 = 0. \tag{4.16}$$

Let $z = \tau(\lambda_2 - \lambda_1)$. Note that z is a real number. The stability of the scheme is determined by whether z falls within the absolute stability region of the BDF-4 method.

Let $z = \tau(\lambda_2 - \lambda_1)$. Note that z is a real number because both λ_1 and λ_2 are real. The stability of the scheme requires that all roots of the characteristic polynomial (4.15) lie within the unit circle $|\xi| \leq 1$. This is equivalent to requiring that the discrete eigenvalue z falls within the absolute stability region S of the BDF-4 method.

Assume that $\lambda_1 \geqslant 0$. Since $\lambda_2 \leqslant 0$ is known, it follows that

$$z = \tau(\lambda_2 - \lambda_1) \leqslant 0. \tag{4.17}$$

In this case, z always lies on the negative real axis. According to Property (i) of Lemma 1, the negative real axis is entirely contained within the stability region S. Therefore, the scheme is unconditionally stable in this regime.

Assume that $\lambda_1 < 0$. Let $Q = |\lambda_1| = -(Q_1 + Q_2) > 0$. The generalized eigenvalue becomes the following:

$$z = \tau(\lambda_2 + Q). \tag{4.18}$$

Since $\lambda_2 \leqslant 0$ and $Q > 0$, if $Q \gg \lambda_2$, then z may become positive. According to Property (ii) of Lemma 1, the BDF-4 method is unstable for large positive real eigenvalues. To ensure stability, z must remain in the stability region, that is to say

$$\begin{aligned} \lambda_2 + Q &\leqslant 0, \\ |\lambda_1| &\leqslant |\lambda_2|. \end{aligned} \tag{4.19}$$

Substituting the expressions for λ_1 and λ_2, we obtain the following stability condition:

$$|Q_1 + Q_2| \leqslant \varepsilon \left| \frac{12(\cos\theta_x - 1)}{h_x^2(5 + \cos\theta_x)} + \frac{12(\cos\theta_y - 1)}{h_y^2(5 + \cos\theta_y)} \right|. \tag{4.20}$$

The overall stability of the scheme in the shock region is dictated by the balance between the numerical dissipation and the gradient growth. The stability condition derived in Eq (4.19) is as follows:

$$|Q| \leqslant \varepsilon \left| \frac{12(\cos\theta_x - 1)}{h_x^2(5 + \cos\theta_x)} + \frac{12(\cos\theta_y - 1)}{h_y^2(5 + \cos\theta_y)} \right|. \tag{4.21}$$

For simplicity, assume uniform grid spacing $h_x = h_y = h$. The inequality can be rewritten as follows:

$$\frac{|Q|h^2}{\varepsilon} \leqslant \left| \frac{12(\cos\theta_x - 1)}{5 + \cos\theta_x} + \frac{12(\cos\theta_y - 1)}{5 + \cos\theta_y} \right|. \tag{4.22}$$

Since the spectral function $S(\theta) = \left|\frac{12(\cos\theta-1)}{5+\cos\theta}\right|$ is bounded for all $\theta \in [-\pi, \pi]$, it follows that

$$\frac{|Q|h^2}{\varepsilon} \leqslant 12. \tag{4.23}$$

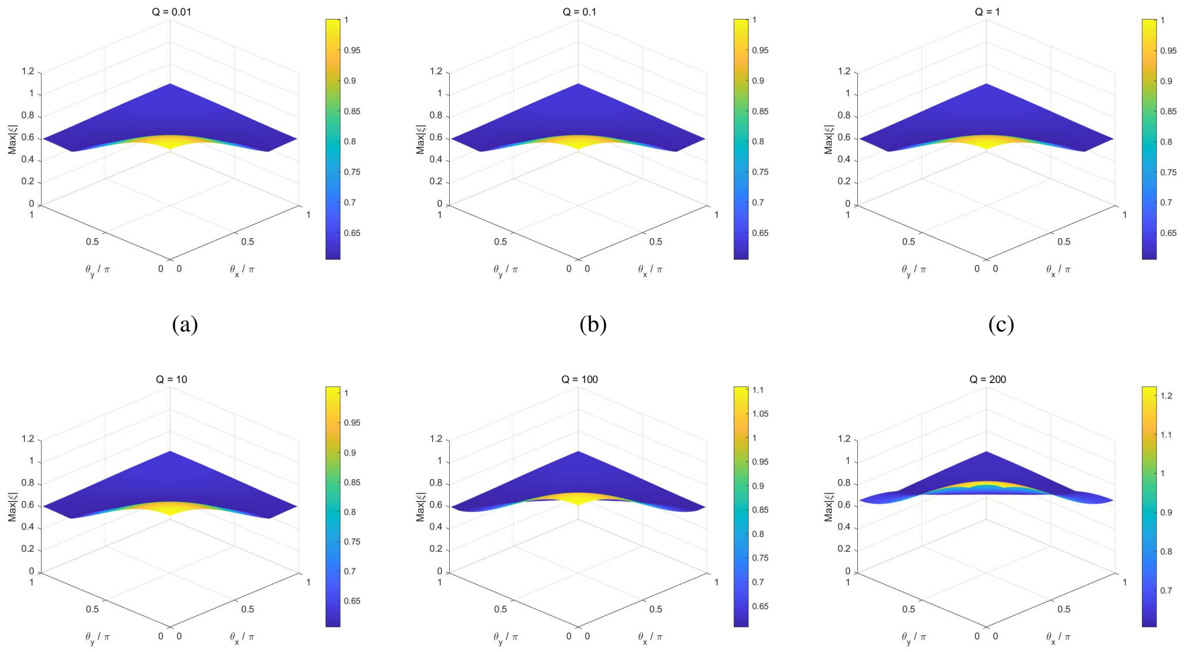
This simplifies to a constraint on the Cell Reynolds Number

$$R = \frac{|Q|h}{\varepsilon} \leqslant \frac{12}{h}, \tag{4.24}$$

or equivalently, a restriction on the grid size h relative to the viscosity and gradients

$$h \leqslant \sqrt{\frac{12\varepsilon}{|Q|}}, \tag{4.25}$$

where $|Q| = |Q_1 + Q_2|$ represents the magnitude of the local gradient. This proves that the scheme is conditionally stable, with the stability dependent on resolving the steep gradients with a sufficiently fine mesh relative to the physical viscosity. □



(a) (b) (c)

(d) (e) (f)

Figure 1. Contour maps under different Q: (a) $Q = 0.01$, (b) $Q = 0.1$, (c) $Q = 1$, (d) $Q = 10$, (e) $Q = 100$, and (f) $Q = 200$.

Figure 1 illustrates the distribution of the maximum amplification factor $|\xi|_{\max}$ under various gradient magnitudes Q. The results for small Q confirm the unconditional stability in smooth regions, while the protrusion at high frequencies for larger Q values visually demonstrates the derived conditional stability constraint. Additionally, the overall dip in the surface highlights the high-frequency dissipation of the BDF-4 scheme, which effectively suppresses numerical oscillations near shocks.

5. Numerical simulation experiments

In this section, the proposed numerical scheme is validated for the nonlinear Burgers' equation by performing a series of numerical tests, demonstrating its reliability and effectiveness. All numerical computations were carried out on a personal computer equipped with an Intel Pentium IV dual-core processor (with a main frequency of 3.0 GHz), using Fortran 77 as the programming language and double-precision floating-point arithmetic algorithms. In this part, we give several numerical experiments to test the various properties of the proposed scheme when solving the system given by Eqs (2.1)–(2.3). The definitions of the L_∞ and L_2 norm errors are given by the following:

$$L_\infty = \max_{i,j} \left| u_{i,j} - U_{i,j} \right|, \qquad L_2 = \sqrt{h^2 \sum_{i=1}^{N_x-1} \sum_{j=1}^{N_y-1} (u_{i,j} - U_{i,j})^2}.$$

The convergence rate is determined by computing the following:

$$Rate = \frac{\ln(error(h_1)/error(h_2))}{\ln(h_1/h_2)},$$

where $error(h_1)$ and $error(h_2)$ are the L_∞ norm errors with grid sizes of h_1 and h_2, respectively. Throughout this paper, the convergence rates are computed based on the L_∞ norm.

5.1. Problem 1

Consider the two-dimensional Burgers' Equation (2.1), both the initial and boundary conditions are prescribed according to the exact solution as follows:

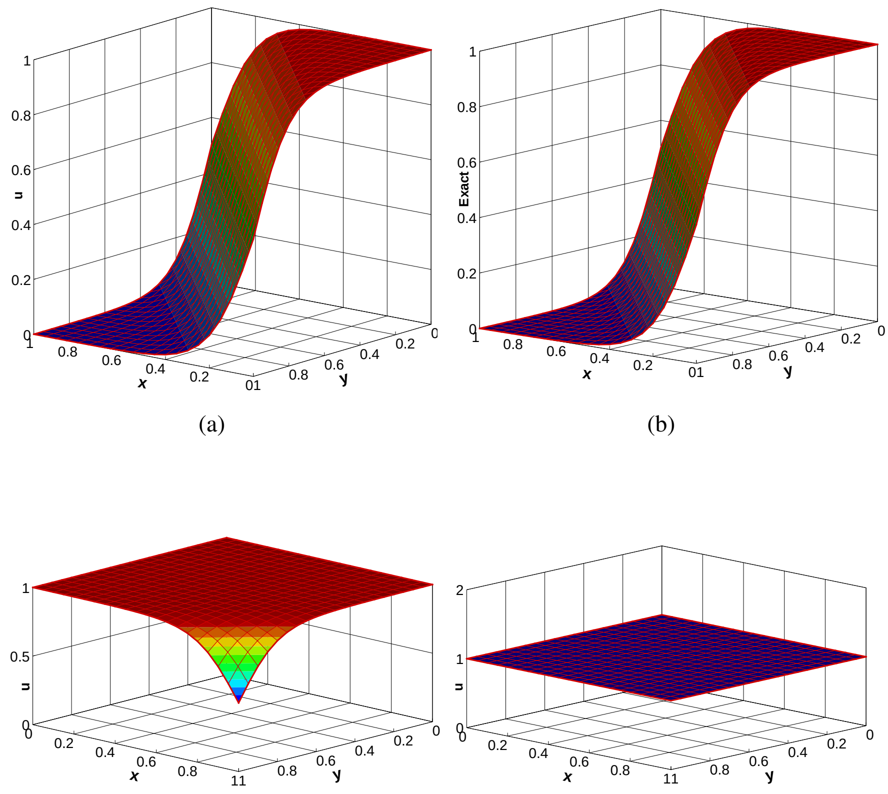
$$u(x, y, t) = 1/(1 + e^{\frac{x+y-t}{2\varepsilon}}), \qquad (x, y, t) \in [0, 1] \times [0, 1] \times [0, T].$$

Table 1 presents the L_∞ norm errors and spatial convergence orders for Problem 1 under different spatial grid steps h for $\tau = h^2/(\sqrt{15}\varepsilon), T = 1/\sqrt{15}$, and ε taking values of 0.1 and 0.01. As shown in Table 1, the LHOC scheme developed here and the HOC approach in [11] reach a fourth-order accuracy, while the method of [13] achieves a sixth-order accuracy, however, the LHOC scheme shows significantly lower errors compared to the sixth-order and HOC schemes in [13] and [11], respectively. This illustrates the LHOC scheme's superior computational accuracy and its excellent performance in both the spatial and temporal directions. Table 2 gives the convergence rate of the scheme in the temporal direction, and the results show that the LHOC scheme also has a fourth-order accuracy in the temporal direction, further verifying the correctness of the theoretical analysis.

Table 3 lists the L_2 and L_∞ norm errors which correspond to different h and τ values when $\varepsilon = 1$ and $T = 1$. By comparing the results of this paper with the sixth-order scheme in [13] and the fourth-order scheme in [29], it can be found that the method in this paper is more accurate, once again verifying the effectiveness and advantages of the high-accuracy spatio-temporal scheme adopted in this paper. Figure 2(a),(b) present the numerical and analytical solutions of Problem 1, respectively. Figure 2(c),(d) show the numerical solutions of Problem 1 at $T = 2$ and $T = 4$, respectively, under the same spatial step $h = 1/20$, time step $\tau = 0.001$ and parameter $\varepsilon = 0.05$. It can be seen from this that there is a good consistency between the exact solution and the numerical solution. Figure 3 presents

Table 1. Comparison of the L_∞ norm error and convergence rate of Problem 1 at $\tau = h^2/\sqrt{15}\varepsilon, T = 1/\sqrt{15}$ for different h and ε.

ε	h	Ref. [13]		Ref. [11]		LHOC	
		L_∞	*Rate*	L_∞	*Rate*	L_∞	*Rate*
0.1	1/20	4.227×10^{-04}	–	1.209×10^{-05}	–	2.076×10^{-06}	–
	1/40	1.333×10^{-05}	4.99	7.552×10^{-07}	3.91	1.394×10^{-07}	3.90
	1/80	2.948×10^{-07}	5.50	4.726×10^{-08}	4.00	8.759×10^{-09}	3.99
	1/160	5.479×10^{-09}	5.75	2.954×10^{-09}	4.00	5.488×10^{-10}	4.00
0.01	1/160	2.937×10^{-02}	–	1.032×10^{-04}	–	1.207×10^{-05}	–
	1/320	9.644×10^{-04}	4.93	6.370×10^{-06}	4.02	6.964×10^{-07}	4.11
	1/640	2.236×10^{-05}	5.43	3.987×10^{-07}	4.00	4.317×10^{-08}	4.01



(a)

(b)

(c)

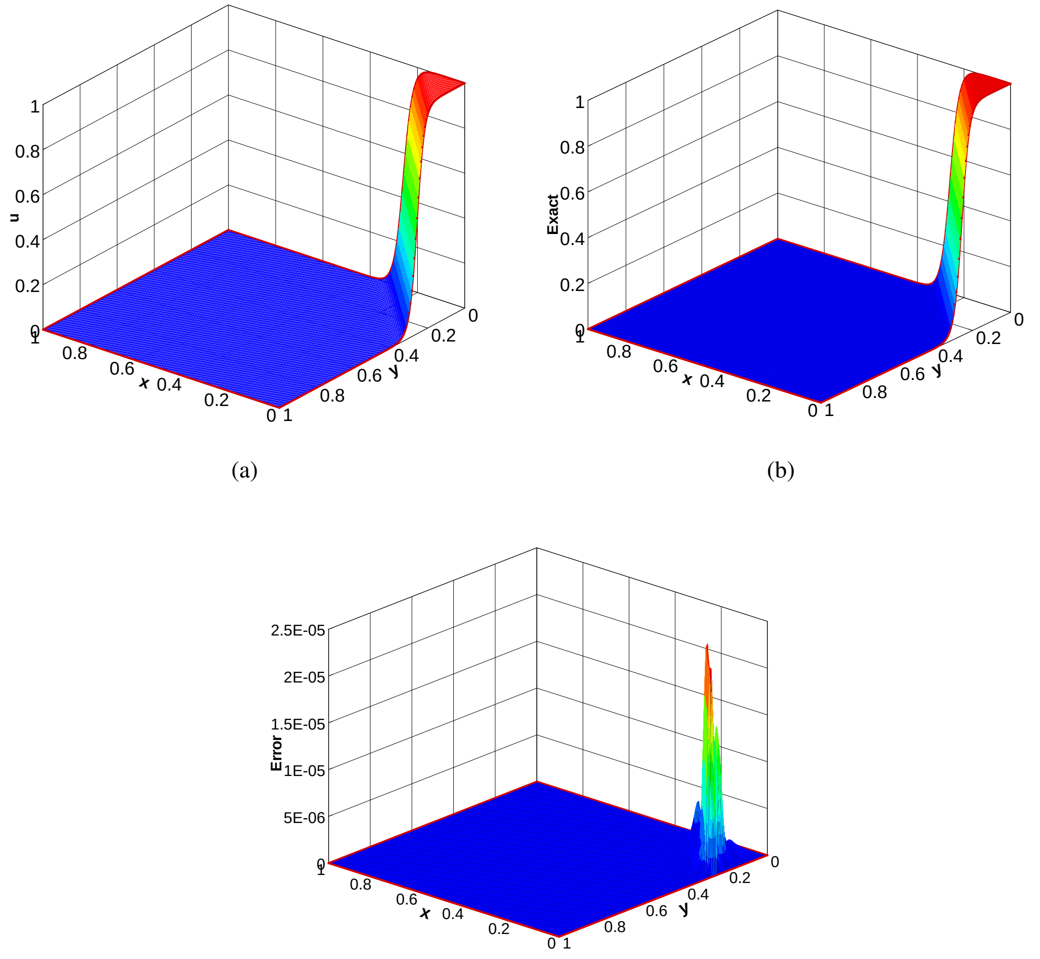
(d)

Figure 2. (a) numerical solutions, (b) exact solutions for $T = 1$ and numerical solutions with (c) $T = 2$, (d) $T = 4$ at the same $\varepsilon = 0.05, h = 1/20, \tau = 0.001$ for Problem 1.

the numerical solution, the exact solution, and the absolute error of Problem 1 under the parameters $h = 1/140, \varepsilon = 0.01, \tau = h^2/\sqrt{15}\varepsilon, T = 1/\sqrt{15}$. The results in the figure show that the method proposed in this paper has a high accuracy.

Table 2. The L_∞ , L_2 norm errors and the rate of convergence with different τ at $T = 1, \varepsilon = 0.1$ and $h = 1/60$ of Problem 1.

τ	L_∞	L_2	*Rate*
0.1	3.154×10^{-02}	6.083×10^{-03}	–
0.05	1.869×10^{-03}	3.586×10^{-04}	4.08
0.025	3.236×10^{-05}	1.313×10^{-05}	5.85
0.0125	1.967×10^{-06}	7.986×10^{-07}	4.04



(a)

(b)

(c)

Figure 3. (a) the numerical solutions, (b) the exact solutions, and (c) the absolute error with $h = 1/140, \varepsilon = 0.01, \tau = h^2/\sqrt{15}\varepsilon, T = 1/\sqrt{15}$ for Problem 1.

5.2. *Problem 2*

Consider the two-dimensional Burgers' Equation (2.1), where the initial and boundary conditions are taken from the exact solution as follows:

$$u(x, y, t) = 0.5 - tanh(\frac{y + x - t}{2\mu}), \qquad (x, y, t) \in [-0.5, 0.5] \times [-0.5, 0.5] \times [0, T].$$

Table 3. Comparison of the L_2 and L_∞ norm errors of Problem 1 at $\varepsilon = 1, T = 0.01$ for different h and τ.

h	τ	Ref. [13]		Ref. [29]		LHOC	
		L_2	L_∞	L_2	L_∞	L_2	L_∞
1/10	0.0005	6.217×10^{-09}	5.907×10^{-09}	6.913×10^{-10}	3.805×10^{-10}	4.343×10^{-11}	5.955×10^{-11}
1/15	0.0001	2.532×10^{-09}	2.188×10^{-09}	1.427×10^{-10}	6.366×10^{-11}	9.712×10^{-12}	1.321×10^{-11}
1/30	0.0001	1.875×10^{-09}	1.010×10^{-09}	1.936×10^{-10}	6.137×10^{-12}	6.486×10^{-13}	8.921×10^{-13}

Table 4. Comparison of the L_2 and L_∞ norm errors of Problem 2 at $T = 1, \varepsilon = 0.1$, and $\tau = 0.001$.

h	Ref. [31]			LHOC		
	L_2	L_∞	*Rate*	L_2	L_∞	*Rate*
1/20	6.872×10^{-02}	4.292×10^{-02}	–	4.352×10^{-06}	4.612×10^{-05}	–
1/40	5.223×10^{-04}	9.254×10^{-04}	5.54	3.143×10^{-07}	3.225×10^{-06}	3.84
1/80	9.123×10^{-06}	4.652×10^{-06}	7.63	1.930×10^{-08}	2.009×10^{-07}	4.01

Table 4 shows the calculated errors and the convergence rate using Time-space Chebyshev Pseudospectral (TCP) scheme in [31] and the LHOC scheme for different h when $T = 1, \varepsilon = 0.1, \tau = 0.001$, and $\mu = 0.1$. The obtained results show that, the LHOC method produces smaller numerical errors than the TCP approach on the tested grid resolutions. These results mainly serve to verify the convergence behavior of the proposed LHOC scheme. Figure 4 presents the numerical solution, the exact solution, and the absolute error of Problem 2 under the parameters $h = 1/40, \varepsilon = 0.05, \tau = 0.001, T = 1$, and $\mu = 0.1$. It is evident from the figure that even for such a problem with sharp variations, the method proposed in this paper still demonstrates an excellent effectiveness. The sharp feature observed near $(0.5, 0.5)$ is not caused by a singularity of the solution, but by a steep interior layer induced by the small ε. The solution remains regular at this point, while its gradient becomes large, thus leading to the observed peak on a finite mesh.

5.3. Problem 3: Coupled Burgers' equation

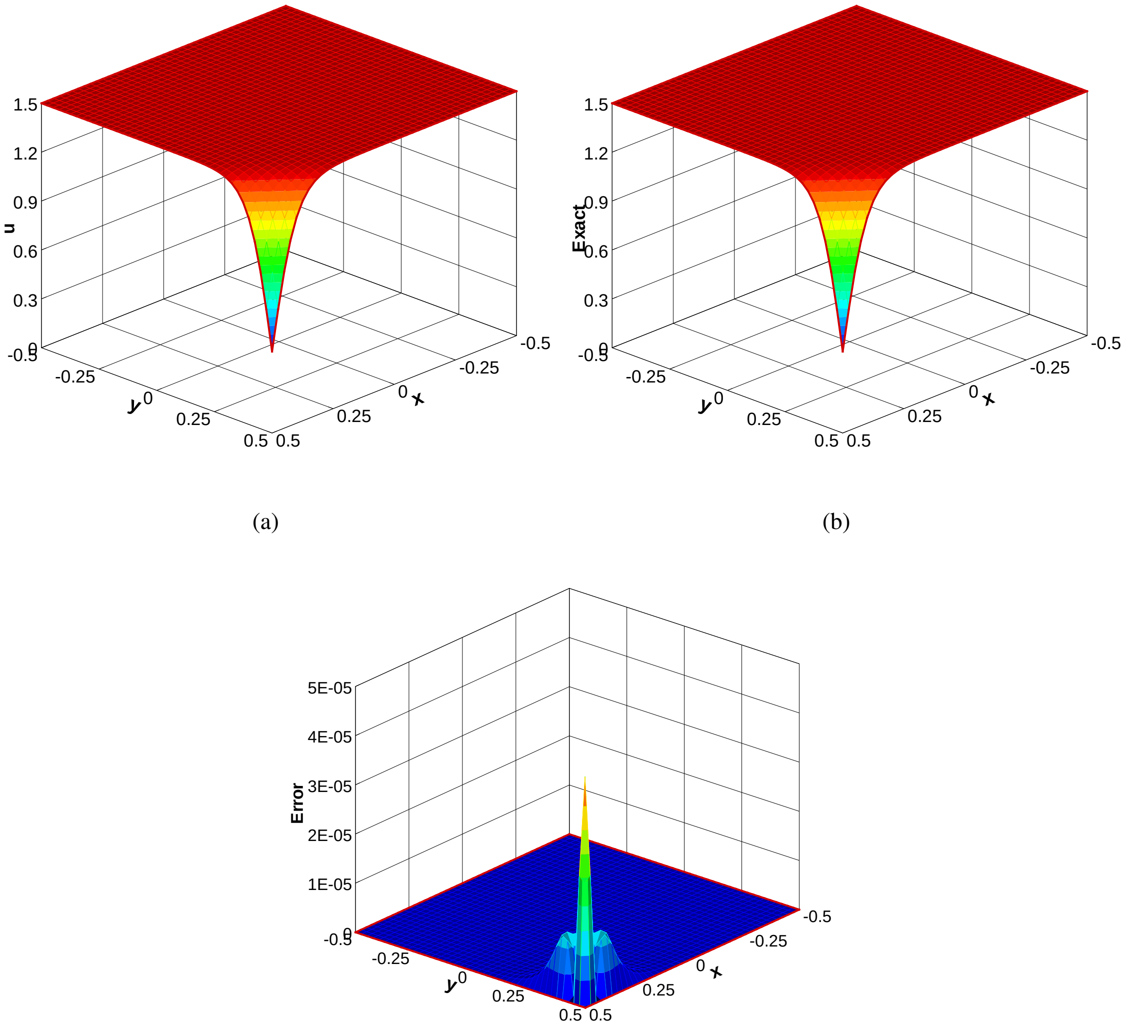
The coupled Burgers' equation is given in the following form:

$$\begin{aligned} &u_t + uu_x + vu_y = \varepsilon(u_{xx} + u_{yy}), \\ &v_t + uv_x + vv_y = \varepsilon(v_{xx} + v_{yy}), \qquad (x, y) \in [0, 1] \times [0, 1], t \in [0, T]. \end{aligned}$$

The exact solution is used to prescribe the initial and boundary conditions as follows:

$$\begin{aligned} u(x, y, t) &= \frac{3}{4} - \frac{1}{4[1 + exp((-4x + 4y - t)/(32\varepsilon))]}, \\ v(x, y, t) &= \frac{3}{4} + \frac{1}{4[1 + exp((-4x + 4y - t)/(32\varepsilon))]}. \end{aligned}$$

To efficiently solve the coupled system, this study employs a decoupled linearization strategy consistent with the proposed scheme. Specifically, by utilizing the four-step extrapolation technique outlined in Eqs (3.9) and (3.10), the nonlinear coupling terms, namely vu_y and uv_x, are explicitly treated as known



(a)

(b)

(c)

Figure 4. (a) the numerical solutions, (b) the exact solutions, (c) the absolute error for Problem 2 with $h = 1/40, \varepsilon = 0.05, \tau = 0.001, T = 1$.

source terms at the current time level. This treatment transforms the original coupled system into two mutually independent linear algebraic systems. Consequently, u and v can be solved separately without the need for inner iterations between variables, thus significantly enhancing the computational efficiency while preserving the fourth-order spatiotemporal accuracy. Table 5 shows the L_2 and L_∞ norm errors. The the coarse grid corresponds to a mesh size of 16, and the HOC scheme in [11] fails to converge since its parameters violate the stability condition. However, and the LHOC scheme is convergent when the mesh number is 16. This shows that the stability of HOC scheme is better than other schemes. Table 6 presents the convergence rate and L_∞ norm error for $\tau = h^2, \varepsilon = 0.01$, and $T = 1$. It can be observed that the proposed LHOC method maintains a fourth-order precision for the coupled system.

When $\varepsilon = 0.01, h = 1/20$, and $\tau = 0.0001$, the absolute errors for $T = 0.01$ and $T = 0.5$ are shown in Table 7, respectively. The LHOC format proposed in this paper is superior in accuracy to the scheme

Table 5. Comparison of the L_2 and L_∞ norm errors for Problem 3 at $\tau = 0.0001, \varepsilon = 0.01$ and $T = 1$.

h	Ref. [11]		LHOC	
	L_2	L_∞	L_2	L_∞
The calculation error of u				
1/16	diverge	diverge	5.991×10^{-05}	3.407×10^{-04}
1/32	1.102×10^{-04}	3.517×10^{-04}	1.704×10^{-06}	7.550×10^{-06}
1/64	7.305×10^{-06}	2.585×10^{-05}	9.877×10^{-08}	3.552×10^{-07}
The calculation error of v				
1/16	diverge	diverge	5.991×10^{-05}	3.407×10^{-04}
1/32	1.102×10^{-04}	3.517×10^{-04}	1.704×10^{-06}	7.550×10^{-06}
1/64	7.305×10^{-06}	2.585×10^{-05}	9.877×10^{-08}	3.552×10^{-07}

Table 6. The L_∞ norm error and the rate of convergence for Problem 3 at $\varepsilon = 0.01, \tau = h^2$ and $T = 1$.

h	Ref. [29]		LHOC	
	L_∞	*Rate*	L_∞	*Rate*
The calculation error of u				
1/16	1.698×10^{-03}	–	3.407×10^{-04}	–
1/32	1.014×10^{-04}	4.02	7.550×10^{-06}	5.50
1/64	7.458×10^{-06}	3.77	3.552×10^{-07}	4.41
1/128	3.466×10^{-07}	4.43	2.173×10^{-08}	4.03
The calculation error of v				
1/16	1.697×10^{-03}	–	3.407×10^{-04}	–
1/32	1.013×10^{-04}	4.02	7.548×10^{-06}	5.50
1/64	7.456×10^{-06}	3.77	3.552×10^{-07}	4.41
1/128	3.466×10^{-07}	4.43	2.173×10^{-08}	4.03

in [11,23]. Table 8 shows numerical solutions and L_2 and L_∞ errors for different points (x, y) when $\varepsilon = 0.01, \tau = 0.0001, h = 1/20$, and $T = 0.01$. As can be seen from the table, compared with [26] and [21], the LHOC scheme is closer to the exact solution. It is not difficult to see that the calculated results in this paper are more reliable.

Under the parameters of $h = 1/20$, $T = 0.5$, $\tau = 0.001$, and $\varepsilon = 0.01$, Figure 5(a),(b) present the numerical solutions of $u(x, y, t)$ and $v(x, y, t)$ in Problem 3, respectively, Figure 5(c),(d) show the corresponding exact solutions, and Figure 5(e),(f) display the absolute error distributions. It can be seen from the figures that the numerical solutions of $u(x, y, t)$ and $v(x, y, t)$ are in good agreement with their exact solutions.

5.4. Problem 4: Coupled Burgers' equation

Consider the following two-dimensional coupled Burgers' equation:

$$u_t + uu_x + vu_y = \varepsilon(u_{xx} + u_{yy}),$$

Table 7. Comparison of the L_∞ errors of Problem 3 for different T at $\tau = 0.0001, \varepsilon = 0.01$ and $h = 1/20$.

T	(x,y)	Ref. [23]		Ref. [11]		LHOC	
		$L_\infty(u)$	$L_\infty(v)$	$L_\infty(u)$	$L_\infty(v)$	$L_\infty(u)$	$L_\infty(v)$
0.01	$(0.1, 0.1)$	5.914×10^{-05}	5.914×10^{-05}	1.220×10^{-05}	1.220×10^{-05}	1.797×10^{-06}	1.880×10^{-06}
	$(0.5, 0.1)$	4.840×10^{-06}	4.840×10^{-06}	5.734×10^{-06}	5.734×10^{-06}	1.651×10^{-08}	1.626×10^{-08}
	$(0.9, 0.1)$	3.419×10^{-08}	3.419×10^{-08}	4.400×10^{-08}	4.400×10^{-08}	1.293×10^{-10}	1.294×10^{-10}
	$(0.3, 0.3)$	5.914×10^{-05}	5.914×10^{-05}	1.136×10^{-05}	1.136×10^{-05}	1.767×10^{-06}	1.767×10^{-06}
	$(0.7, 0.3)$	4.840×10^{-06}	4.840×10^{-06}	5.756×10^{-06}	5.756×10^{-06}	4.743×10^{-08}	4.743×10^{-08}
	$(0.1, 0.5)$	1.643×10^{-06}	1.643×10^{-06}	5.503×10^{-06}	5.503×10^{-06}	6.997×10^{-08}	7.027×10^{-08}
	$(0.5, 0.5)$	5.914×10^{-05}	5.914×10^{-05}	1.136×10^{-05}	1.136×10^{-05}	1.768×10^{-06}	1.768×10^{-06}
0.5	$(0.1, 0.1)$	2.777×10^{-04}	2.777×10^{-04}	6.096×10^{-04}	6.096×10^{-04}	7.511×10^{-05}	7.511×10^{-05}
	$(0.5, 0.1)$	4.521×10^{-04}	4.521×10^{-04}	1.798×10^{-05}	1.798×10^{-05}	1.066×10^{-06}	1.066×10^{-06}
	$(0.9, 0.1)$	3.374×10^{-06}	3.374×10^{-06}	1.117×10^{-07}	1.117×10^{-07}	4.729×10^{-08}	4.729×10^{-08}
	$(0.3, 0.3)$	2.777×10^{-04}	2.777×10^{-04}	1.388×10^{-03}	1.388×10^{-03}	1.179×10^{-05}	1.179×10^{-05}
	$(0.7, 0.3)$	4.521×10^{-04}	4.521×10^{-04}	5.022×10^{-05}	5.022×10^{-05}	7.260×10^{-07}	7.260×10^{-07}
	$(0.1, 0.5)$	2.866×10^{-04}	2.866×10^{-04}	1.299×10^{-04}	1.299×10^{-04}	1.001×10^{-05}	1.001×10^{-05}
	$(0.5, 0.5)$	2.777×10^{-04}	2.777×10^{-04}	1.550×10^{-03}	1.550×10^{-03}	2.345×10^{-05}	2.345×10^{-05}

Table 8. Comparison between numerical solutions $u(x, y, t)$ and $v(x, y, t)$ of the proposed method with other existing methods of Problem 3 with $\varepsilon = 0.01, \tau = 0.0001, h = 1/20$ and $T = 0.01$.

(x,y)	Ref. [26]		Ref. [21]		LHOC		Exact	
	$u(x,y,t)$	$v(x,y,t)$	$u(x,y,t)$	$v(x,y,t)$	$u(x,y,t)$	$v(x,y,t)$	$u(x,y,t)$	$v(x,y,t)$
$(0.1, 0.1)$	0.62310	0.87688	0.62305	0.87695	0.623049	0.876951	0.623048	0.876952
$(0.5, 0.1)$	0.50161	0.99837	0.50162	0.99838	0.501622	0.998378	0.501615	0.998383
$(0.9, 0.1)$	0.50000	0.99998	0.50001	0.99999	0.500011	0.999989	0.500007	0.999998
$(0.3, 0.3)$	0.62311	0.87689	0.62305	0.87695	0.623049	0.876951	0.623056	0.876956
$(0.7, 0.3)$	0.50162	0.99838	0.50162	0.99838	0.501622	0.998378	0.501624	0.998385
$(0.1, 0.5)$	0.74827	0.75172	0.74827	0.75172	0.748274	0.751726	0.748272	0.751733
$(0.5, 0.5)$	0.62311	0.87689	0.62305	0.87695	0.623049	0.876951	0.623051	0.876952
$(0.9, 0.5)$	0.50162	0.99838	0.50162	0.99838	0.501622	0.998378	0.501615	0.998381
$(0.3, 0.7)$	0.74827	0.75173	0.74827	0.75173	0.748274	0.751726	0.748268	0.751734
$(0.7, 0.7)$	0.62311	0.87689	0.62305	0.87695	0.623049	0.876951	0.623046	0.876954
$(0.1, 0.9)$	0.74998	0.75001	0.74999	0.75001	0.749988	0.750012	0.749987	0.750009
$(0.5, 0.9)$	0.74827	0.75173	0.74827	0.75172	0.748274	0.751726	0.748271	0.751728
$(0.9, 0.9)$	0.62311	0.87689	0.62308	0.87695	0.623050	0.876950	0.623052	0.876946
L_∞					1.635×10^{-05}	1.620×10^{-05}		
L_2					2.343×10^{-06}	2.343×10^{-06}		

$$v_t + uv_x + vv_y = \varepsilon(v_{xx} + v_{yy}).$$

The exact solution is as follows:

$$\begin{aligned} u(x,y,t) &= \frac{x+y-2xt}{1-2t^2}, \\ v(x,y,t) &= \frac{x+y-2yt}{1-2t^2}, \qquad (x,y) \in [0,0.5]\times[0,0.5], t \in [0,T]. \end{aligned}$$

Table 9. Comparison of the absolute errors of Problem 4 at $\tau = 0.0001, \varepsilon = 0.01, h = 1/20$ and $T = 0.1$.

(x,y)	Ref. [23]		Ref. [11]		LHOC	
	$L_\infty(u)$	$L_\infty(v)$	$L_\infty(u)$	$L_\infty(v)$	$L_\infty(u)$	$L_\infty(v)$
$(0.1,0.1)$	3.308×10^{-06}	1.054×10^{-06}	9.400×10^{-07}	7.094×10^{-07}	5.028×10^{-13}	8.820×10^{-13}
$(0.3,0.1)$	5.562×10^{-06}	3.308×10^{-06}	5.425×10^{-07}	2.589×10^{-06}	4.222×10^{-13}	4.594×10^{-13}
$(0.2,0.2)$	6.615×10^{-06}	2.108×10^{-06}	1.902×10^{-06}	1.428×10^{-06}	5.492×10^{-13}	8.808×10^{-13}
$(0.4,0.2)$	8.869×10^{-06}	2.254×10^{-06}	1.488×10^{-06}	3.350×10^{-06}	2.798×10^{-14}	3.042×10^{-14}
$(0.1,0.3)$	7.669×10^{-06}	7.523×10^{-06}	3.183×10^{-06}	2.330×10^{-07}	6.778×10^{-14}	1.226×10^{-13}
$(0.3,0.3)$	9.923×10^{-06}	3.162×10^{-06}	2.853×10^{-06}	2.141×10^{-06}	8.216×10^{-15}	1.255×10^{-14}
$(0.2,0.4)$	1.098×10^{-05}	8.577×10^{-06}	4.108×10^{-06}	9.133×10^{-07}	1.199×10^{-14}	8.882×10^{-16}
$(0.3,0.4)$	1.210×10^{-05}	6.396×10^{-06}	3.940×10^{-06}	1.865×10^{-06}	1.110×10^{-16}	8.132×10^{-15}

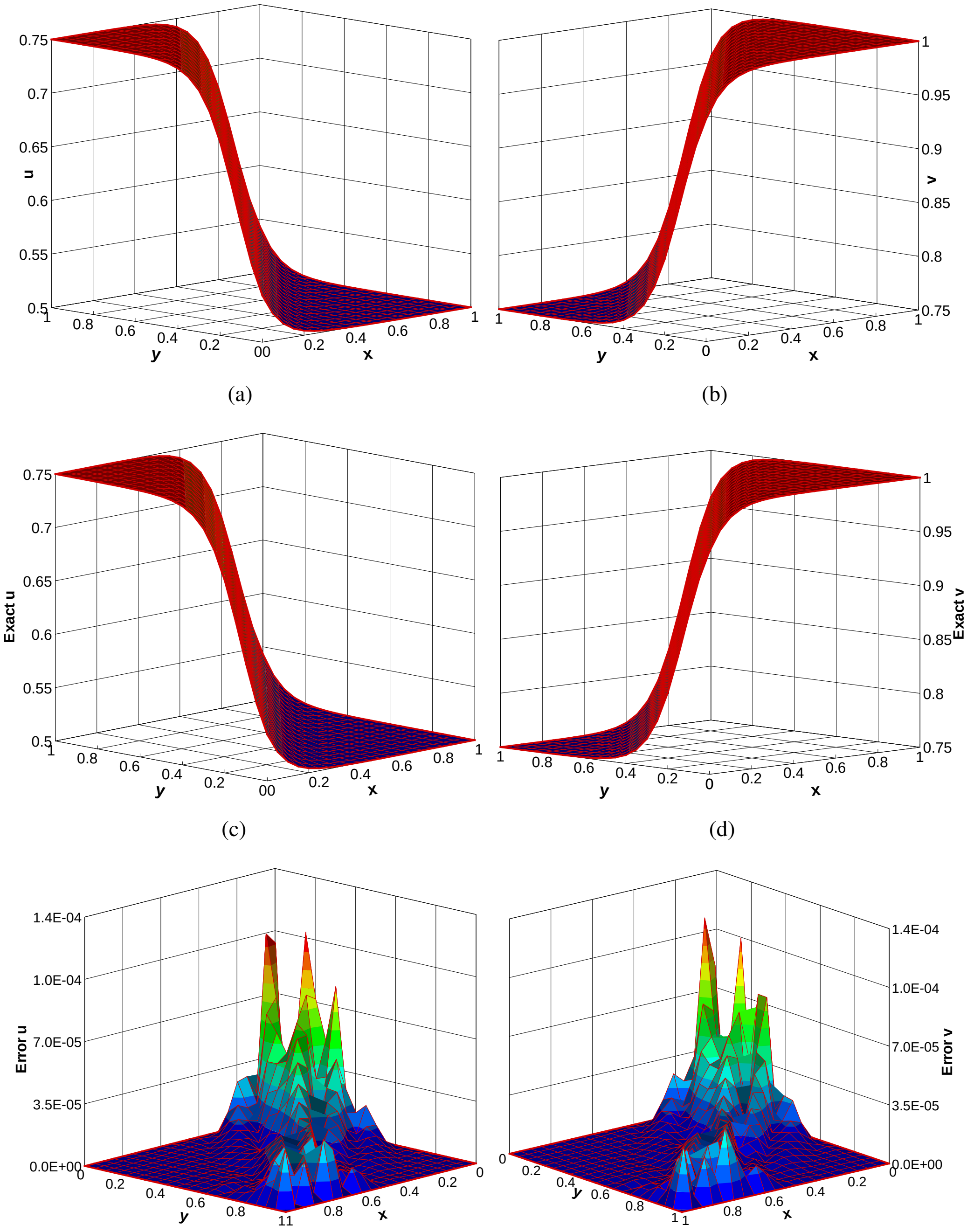
Table 10. Comparison of the absolute errors of Problem 4 at $h = 1/20$, $\tau = 0.0001, \varepsilon = 0.01$ and $T = 0.4$.

(x,y)	Ref. [23]		Ref. [29]		LHOC	
	$L_\infty(u)$	$L_\infty(v)$	$L_\infty(u)$	$L_\infty(v)$	$L_\infty(u)$	$L_\infty(v)$
$(0.1,0.1)$	1.019×10^{-04}	3.548×10^{-04}	6.515×10^{-06}	1.392×10^{-06}	8.530×10^{-13}	4.560×10^{-12}
$(0.3,0.1)$	5.587×10^{-04}	1.019×10^{-04}	4.858×10^{-07}	7.145×10^{-06}	2.218×10^{-14}	2.273×10^{-12}
$(0.2,0.2)$	2.039×10^{-04}	7.097×10^{-04}	1.572×10^{-05}	4.253×10^{-06}	8.181×10^{-13}	4.158×10^{-12}
$(0.4,0.2)$	6.607×10^{-04}	4.568×10^{-04}	5.972×10^{-06}	8.558×10^{-06}	2.461×10^{-13}	8.115×10^{-13}
$(0.1,0.3)$	1.509×10^{-04}	1.317×10^{-03}	2.050×10^{-05}	9.647×10^{-06}	1.142×10^{-13}	3.154×10^{-13}
$(0.3,0.3)$	3.058×10^{-04}	1.065×10^{-03}	2.255×10^{-05}	6.272×10^{-06}	1.510×10^{-13}	6.021×10^{-13}
$(0.2,0.4)$	4.900×10^{-05}	1.672×10^{-03}	1.995×10^{-05}	8.400×10^{-06}	5.984×10^{-14}	3.098×10^{-14}
$(0.3,0.4)$	1.794×10^{-04}	1.546×10^{-03}	2.022×10^{-05}	7.193×10^{-06}	5.995×10^{-14}	4.852×10^{-14}

Tables 9 and 10 present the absolute errors at different spatial points under the conditions for Problem 4. The data in Tables 9 and 10 indicate that the LHOC scheme proposed in this paper has a higher computational accuracy, with the error being reduced by approximately 10 orders of magnitude compared to the methods in the literature. Under the parameter conditions of $h = 1/20$, $T = 1$, $\tau = 0.0001$, and $\varepsilon = 0.01$, Figure 6(a),(b) present the numerical solutions of $u(x,y,t)$ and $v(x,y,t)$ in Problem 4, respectively, while Figure 6(c),(d) show the corresponding absolute error distributions.

6. Conclusions

This paper constructed a LHOC finite difference scheme to solve the two-dimensional nonlinear Burgers' equation. The scheme has a fourth-order accuracy in both the time and space directions.



(a)

(b)

(c)

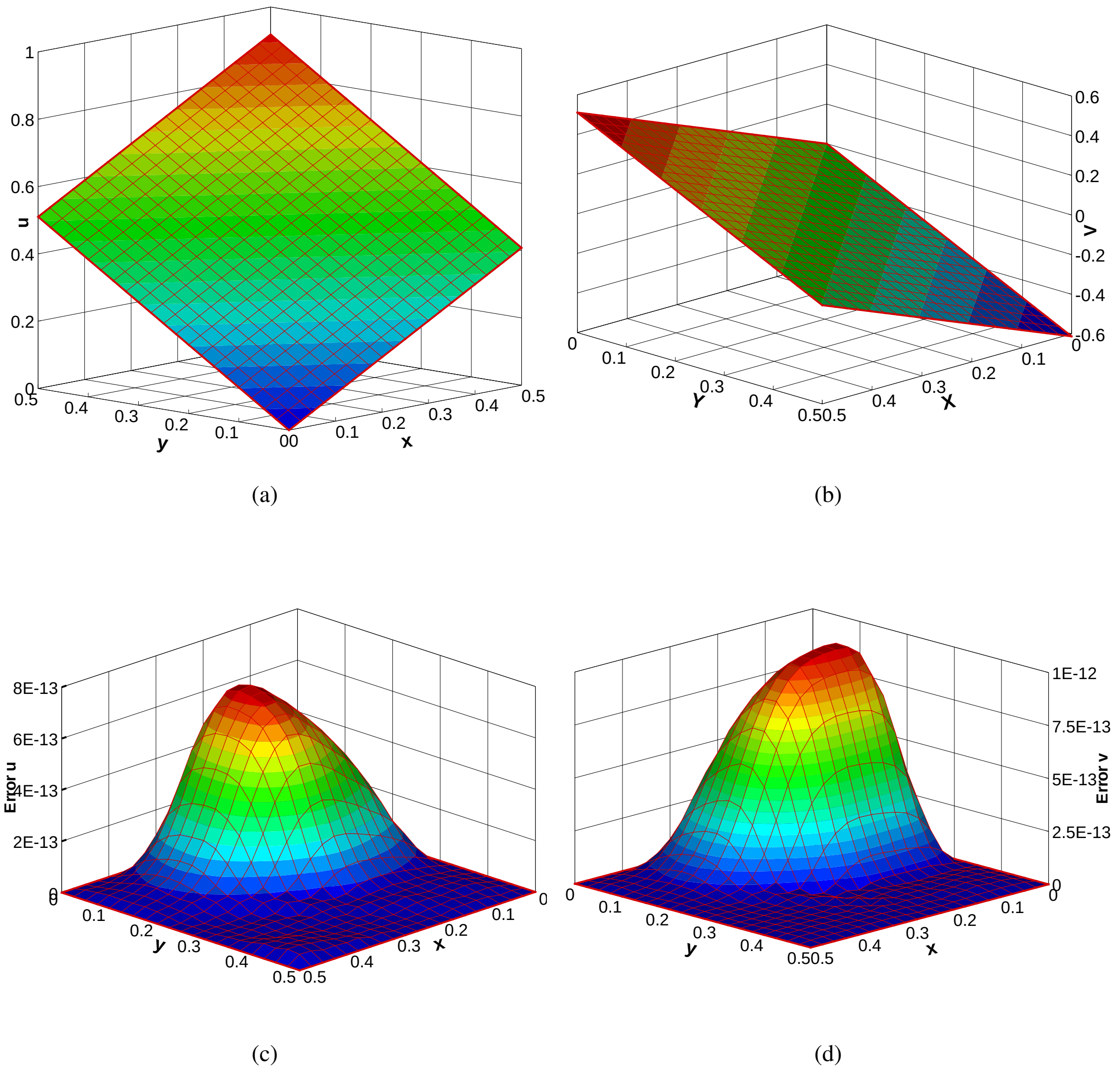
(d)

(e)

(f)

Figure 5. When $h = 1/20, T = 0.5, \tau = 0.001, \varepsilon = 0.01$, (a) the numerical solutions of $u(x, y, t)$, (b) the numerical solutions of $v(x, y, t)$, (c) the exact solutions of $u(x, y, t)$, (d) the exact solutions of $v(x, y, t)$, (e) the absolute error of $u(x, y, t)$, (f) the absolute error of $v(x, y, t)$ for Problem 3.

In addition to numerical verification, a rigorous theoretical analysis established that the scheme is conditionally uniquely solvable and conditionally stable, thus providing a solid mathematical foundation for its reliability. A series of numerical experiments verified the excellent performance of the scheme



(a)

(b)

(c)

(d)

Figure 6. When $T = 0.1, h = 1/20, \varepsilon = 0.01$, and $\tau = 0.0001$, the numerical solutions of (a) $u(x, y, t)$ and (b) $v(x, y, t)$, the absolute error of (c) $u(x, y, t)$ and (d) $v(x, y, t)$ for Problem 4.

in terms of the computational accuracy, convergence order, and numerical stability, and the results were compared with those in the existing literature. Overall, compared with the existing numerical methods to solve such problems, the proposed method has three significant advantages, First, the high-order compact scheme ensures strong spatial compactness, thereby effectively mitigating numerical oscillations and enhancing solution stability in high-frequency or complex propagation scenarios. By restricting operations to neighboring nodes, it adopts a localized formulation that reduces the computational cost and improves the efficiency. Second, the scheme attains high-order temporal accuracy, thereby outperforming many existing approaches under comparable conditions, and maintains robustness even at relatively large time steps. Third, the proposed method is a linear scheme and has a high computational efficiency. In future research, we plan to explore the potential of this method at the application level for practical physical systems, such as turbulence simulations, and to verify its universality and adaptability across complex engineering problems. Through systematic theoretical improvements and application expansion, it is expected to further enhance the computational performance of the method, broaden its application scope, and provide more efficient, stable, and reliable numerical simulation tools for complex scientific and engineering problems.

Use of AI tools declaration

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

Acknowledgments

This work is partially supported by Foreign Expert Program (G2023045002L), Natural Science Foundation of Ningxia (2025AAC030584), 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), Plan for Introducing Foreign Talents in Ningxia.

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. K. T. Elgindy, High-order, stable, and efficient pseudospectral method using barycentric Gegenbauer quadratures, *Appl. Numer. Math.*, **113** (2017), 1–25. https://doi.org/10.1016/j.apnum.2016.10.014
2. K. Fujisawa, A. Asada, Nonlinear parametric sound enhancement through different fluid layer and its application to noninvasive measurement, *Measurement*, **94** (2016), 726–733. https://doi.org/10.1016/j.measurement.2016.09.004
3. Y. D. Jin, J. Zhou, Z. K. Shi, Lattice hydrodynamic model for traffic flow on curved road with passing, *Nonlinear Dyn.*, **89** (2017), 107–124. https://doi.org/10.1007/s11071-017-3439-8
4. K. Shah, Solution of Burgers' equation in a one-dimensional groundwater recharge by spreading using q-Homotopy analysis method, *Eur. J. Pure Appl. Math.*, **9** (2016), 114–124. https://www.ejpam.com/index.php/ejpam
5. L. Yang, X. Pu, Derivation of the Burgers' equation from the gas dynamics, *Commun. Math. Sci.*, **14** (2016), 671–682. https://doi.org/10.4310/cms.2016.v14.n3.a4
6. A. Khoshfetrat, M. Abedini, A hybrid DQ/LMQRBF-DQ approach for numerical solution of Poisson-type and Burger's equations in irregular domain, *Appl. Math. Modell.*, **36** (2012), 1885–1901. https://doi.org/10.1016/j.apm.2011.07.079
7. R. Mohanty, N. Setia, A new high accuracy two-level implicit off-step discretization for the system of two space dimensional quasi-linear parabolic partial differential equations, *Appl. Math. Comput.*, **219** (2012), 2680–2697. https://doi.org/10.1016/j.amc.2012.08.100
8. M. Seydaoğlu, U. Erdoğan, T. Öziş, Numerical solution of Burgers' equation with high order splitting methods, *J. Comput. Appl. Math.*, **291** (2016), 410–421. https://doi.org/10.1016/j.cam.2015.04.021

9. R. C. Mittal, R. Rohila, A study of one dimensional nonlinear diffusion equations by Bernstein polynomial based differential quadrature method, *J. Math. Chem.*, **55** (2017), 673–695. https://doi.org/10.1007/s10910-016-0703-y

10. I. A. Ganaie, V. K. Kukreja, Numerical solution of Burgers' equation by cubic Hermite collocation method, *Appl. Math. Comput.*, **237** (2014), 571–581. https://doi.org/10.1016/j.amc.2014.03.102

11. X. J. Yang, Y. B. Ge, L. Zhang, A class of high-order compact difference schemes for solving the Burgers' equations, *Appl. Math. Comput.*, **358** (2019), 394–417. https://doi.org/10.1016/j.amc.2019.04.023

12. V. Mukundan, A. Awasthi, Linearized implicit numerical method for Burgers' equation, *Nonlinear Eng.*, **5** (2016), 219–234. https://doi.org/10.1515/nleng-2016-0031

13. T. Zhanlav, O. Chuluunbaatar, V. Ulziibayar, Higher-order accurate numerical solution of unsteady Burgers' equation, *Appl. Math. Comput.*, **250** (2015), 701–707. https://doi.org/10.1016/j.amc.2014.11.013

14. X. Miao, J. Gan, T. Ma, L. Wu, A linear time-space high-order compact finite difference scheme for solving the one-dimensional nonlinear Burgers' equation, *J. Appl. Anal. Comput.*, **15** (2025), 3226–3245. https://doi.org/10.11948/20240527

15. A. Korkmaz, I. Dag, Cubic B-spline differential quadrature methods and stability for Burgers' equation, *Eng. Comput.*, **30** (2013), 320–344. https://doi.org/10.1108/02644401311314312

16. G. Arora, B. Singh, Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method, *Appl. Math. Comput.*, **224** (2013), 166–177. https://doi.org/10.1016/j.amc.2013.08.071

17. H. Shukla, M. Tamsir, V. Srivastava, J. Kumar, Numerical solution of two dimensional coupled viscous Burgers' equation using modified cubic B-spline differential quadrature method, *AIP Adv.*, **4** (2014), 117134. https://doi.org/10.1063/1.4902507

18. R. Mittal, A. Tripathi, A. Ali, Numerical solutions of two-dimensional Burgers' equations using modified B-cubic B-spline finite elements, *Eng. Comput.*, **32** (2015), 1275–1306. https://doi.org/10.1108/ec-04-2014-0067

19. R. C. Mittal, R. K. Jain, Numerical solutions of nonlinear Burgers' equation with modified cubic b-splines collocation method, *Appl. Math. Comput.*, **218** (2012), 7839–7855. https://doi.org/10.1016/j.amc.2012.01.059

20. C. Fan, P. Li, Generalized finite difference method for solving two-dimensional Burgers' equations, *Procedia Eng.*, **79** (2014), 55–60. https://doi.org/10.1016/j.proeng.2014.06.310

21. R. C. Mittal, R. Jiwari, Differential quadrature method for two-dimensional Burgers' equations, *Int. J. Comput. Methods Eng. Sci. Mech.*, **10** (2009), 450–459. https://doi.org/10.1080/15502280903111424

22. A. A. Soliman, On the solution of two-dimensional coupled Burgers' equations by variational iteration method, *Chaos, Solitons Fractals*, **40** (2009), 1146–1155. https://doi.org/10.1016/j.chaos.2007.08.069

23. H. Zhu, H. Shu, M. Ding, Numerical solutions of two-dimensional Burgers' equations by discrete adomian decomposition method, *Comput. Math. Appl.*, **60** (2010), 840–848. https://doi.org/10.1016/j.camwa.2010.05.031

24. G. Zhao, X. Yu, R. Zhang, The new numerical method for solving the system of two-dimensional Burgers' equations, *Comput. Math. Appl.*, **62** (2011), 3279–3291. https://doi.org/10.1016/j.camwa.2011.08.044

25. S. Baharlouei, R. Mokhtari, N. Chegini, Solving two-dimensional coupled Burgers' equations via a stable hybridized discontinuous Galerkin method, *Iran. J. Numer. Anal. Optim.*, **13** (2023), 397–425. https://doi.org/10.22067/ijnao.2023.80916.1215

26. A. R. Bahadır, A fully implicit finite-difference scheme for two-dimensional Burgers equations, *Appl. Math. Comput.*, **137** (2003), 131–137. https://doi.org/10.1016/s0096-3003(02)00091-7

27. X. Chen, Y. Chen, A weak Galerkin finite element method for the two-dimensional Burgers' equation, *Comput. Math. Appl.*, **196** (2025), 84–103. https://doi.org/10.1016/j.camwa.2025.07.010

28. S. Gottlieb, C. Wang, Stability and convergence analysis of fully discrete Fourier collocation spectral method for 3-D viscous Burgers' equation, *J. Sci. Comput.*, **53** (2012), 102–128. https://doi.org/10.1007/s10915-012-9621-8

29. X. J. Yang, Y. B. Ge, B. Lan, A class of compact finite difference schemes for solving the 2D and 3D Burgers' equations, *Math. Comput. Simul.*, **185** (2021), 510–524. https://doi.org/10.1016/j.matcom.2021.01.009

30. X. Peng, D. Xu, W. Qiu, Pointwise error estimates of compact difference scheme for mixed-type time-fractional Burgers' equation, *Math. Comput. Simul.*, **208** (2023), 702–726. https://doi.org/10.1016/j.matcom.2023.02.004

31. A. K. Mittal, A space-time pseudospectral method for solving multi-dimensional quasi-linear parabolic partial differential (Burgers') equations, *Appl. Numer. Math.*, **195** (2024), 39–53. https://doi.org/10.1016/j.apnum.2023.09.005

32. Y. Sun, Q. Chen, T. Chen, L. Yong, Explicit numerical manifold characteristic Galerkin method for solving Burgers' equation, *Axioms*, **13** (2024), 343. https://doi.org/10.3390/axioms13060343

33. X. Zhang, Y. Zhang, Q. Jin, Y. Jin, Two sixth-order, L^∞ convergent, and stable compact difference schemes for the generalized Rosenau-KdV-RLW equation, *J. Comput. Appl. Math.*, **459** (2025), 116382. https://doi.org/10.1016/j.cam.2024.116382

34. A. Polwang, K. Poochinapan, B. Wongsaijai, Numerical simulation of wave flow : Integrating the BBM-KdV equation using compact difference schemes, *Math. Comput. Simul.*, **236** (2025), 70–89. https://doi.org/10.1016/j.matcom.2025.03.012

35. S. D. Wang, T. F. Ma, L. L. Wu, X. J. Yang, Two high-order compact finite difference schemes for solving the nonlinear generalized Benjamin-Bona-Mahony-Burgers equation, *Appl. Math. Comput.*, **496** (2025), 129360. https://doi.org/10.1016/j.amc.2025.129360

36. S. Lele, Compact finite difference schemes with spectral-like resolution, *J. Comput. Phys.*, **103** (1992), 16–42. https://doi.org/10.1016/0021-9991(92)90324-r

37. Y. Kang, H. Liao, Energy stability of BDF methods up to fifth-order for the molecular beam epitaxial model without slope selection, *J. Sci. Comput.*, **91** (2022), 47. https://doi.org/10.1007/s10915-022-01830-x