*Electronic Research Archive*

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

ERA, 33(3): 1490–1509.
DOI: 10.3934/era.2025070
Received: 19 January 2025
Revised: 02 March 2025
Accepted: 06 March 2025
Published: 17 March 2025

*Research article*

# Numerical solution of a coupled Burgers' equation via barycentric interpolation collocation method

**Xiumin Lyu**[1], **Jin Li**[2,*] **and Wanjun Song**[2]

[1] School of Science, Shandong Jiaotong University, Jinan 250357, China
[2] School of Science, Shandong Jianzhu University, Jinan 250101, China

* **Correspondence:** Email: lijin@lsec.cc.ac.cn.

**Abstract:** In this study, the numerical solution of a two-dimensional coupled Burgers' equation is investigated by the barycentric interpolation collocation method. The spatial-temporal domain of the equation is made discrete by employing barycentric interpolation, and a corresponding differentiation matrix based on this interpolation technique is constructed. The effectiveness and high accuracy of the proposed method are demonstrated for solving the two-dimensional coupled Burgers' equation.

**Keywords:** numerical calculation; barycentric collocation method; Burgers' equation; error distribution

## 1. Introduction

Nonlinear partial differential equations are widely encountered in various scientific and engineering fields, including mathematical biology, plasma physics, chemical physics, and fluid dynamics. Burgers' equation, a prominent partial differential equation, is extensively utilized in modeling viscous fluid flow, shock waves, shallow water waves, and traffic flow, etc. [1, 2]. It also serves a benchmark in testing various numerical algorithms. The resolution of this equation is crucial for the development of mathematical models and achieving a comprehensive understanding of physical phenomena. Numerous mathematicians and researchers are exploring a variety of methods, including not only numerical techniques [3, 4], such as the differential quadrature method [5] and the finite-difference method [6], but also analytical approaches [7–10], such as the decomposition method [11] and the homotopy perturbation method [12], to deepen their understanding of the dynamics and behavior of the model. For example, Kumar and Pandit [4] introduced a composite numerical format for solving the one-dimensional (1D) coupled Burgers' equation by finite difference methods and Haar wavelets. Soliman [7] applied the variational iteration method to derive accurate solutions for both the Burgers' equation and the coupled Burgers' equation, while Kaya [11] obtained exact solutions for the coupled

Burgers' equation through the Adomian decomposition method. The coupled Burgers' equation is often used to simulate various coupled transport phenomena. However, due to its nonlinear nature, analytical solution for the coupled Burgers' equation can not be obtained easily. As a result, numerical solutions for this type of equation remain a vital method for investigating its dynamic behavior.

In this paper, we examine a two-dimensional (2D) coupled nonlinear Burgers' equation [13, 14]

$$\begin{cases} u_t + uu_x + vu_y - r(u_{xx} + u_{yy}) = 0, \\ v_t + uv_x + vv_y - r(v_{xx} + v_{yy}) = 0, \end{cases} \tag{1.1}$$

where $q_t = \partial q/\partial t$, $q_x = \partial q/\partial x$, $q_y = \partial q/\partial y$, $q_{xx} = \partial^2 q/\partial x^2$, and $q_{yy} = \partial^2 q/\partial y^2$ $(q = u, v)$. Furthermore, the variables $u$ and $v$ denote the velocity components that require determination, and $r = 1/R$, where $R$ is the Reynolds number. The initial conditions of Eq (1.1) is denoted as

$$u(x, y, 0) = f_0(x, y), \quad v(x, y, 0) = g_0(x, y), \quad (x, y) \in D,$$

where $D = [a, b] \times [c, d]$, $f_0, g_0$ are given functions. In addition, the boundary conditions of Eq (1.1) are denoted as

$$u(x, y, t) = f(x, y, t), \quad v(x, y, t) = g(x, y, t), \quad x, y \in \partial D,$$

where $t > 0$, $\partial D$ denotes the boundary of region $D$, and $f, g$ are all known functions. This coupled viscous Burgers' equation offers an intuitive and straightforward model for investigating the sedimentation or scaling volume concentration evolution of two types of particles in fluid suspensions or colloids under the influence of gravitational forces.

The analysis of a coupled Burgers' equation is instrumental in elucidating the convection–diffusion phenomenon, which is influenced by the viscosity coefficients associated with the nonlinear convection and diffusion terms. Hopf [15] and Cole [16] have demonstrated that the 1D Burgers' equation can be effectively addressed by transforming it into a linear homogeneous heat equation, with solutions represented through Fourier series. In [17], the exact solution of the 2D coupled Burgers' equation could generally be obtained only under specific initial and boundary conditions. Furthermore, the inclusion of nonlinear convection terms and viscous parameters complicates the numerical analysis of the viscous Burgers' equation, making it a consistently challenging task. In the past few years, several significant numerical approaches have been developed for the coupled Burgers' equations. For instance, Chai and Ouyang [18] explored methods for reducing shock oscillations, including the streamline upwind/Petrov-Galerkin method and other related techniques. The authors of [19] introduced a weak Galerkin finite element method for solving the 2D coupled Burgers' equation. A novel fourth-order compact difference scheme is proposed in [20], along with its application to solving the mixed-type time-fractional Burgers' equation. The work in [21] presented a new numerical method for addressing the coupled Burgers' equation, utilizing the collocation finite element method with cubic trigonometric and quintic B-splines as approximation functions. The performance and accuracy of this method are demonstrated through the calculation of the error norms $L_2$ and $L_\infty$. Another study [22] examined the application of the RBF-QR method in obtaining numerical solutions for the 1D coupled viscous Burgers' equation. Lastly, Park et al. [23] developed an algorithm designed to reduce the computational cost associated with the backward semi-Lagrangian method (BSLM) for solving nonlinear convection-diffusion equations.

In recent years, the barycentric interpolation method has gained prominence due to its robust node adaptability and excellent numerical stability for both special and equidistant node distributions. Consequently, a substantial body of literature has employed this method to address various numerical challenges. Notable applications include Volterra integro-differential equations [24], nonlinear high-dimensional Fredholm integral equations [25], the heat conduction equation [26], plane elasticity problems [27], and semi-infinite domain problems [28], and so on. Furthermore, Li and Wang [29] and Wang and Li [30] have successfully applied the barycentric interpolation method to a wide range of practical engineering issues, including dynamic diffusion problems, vibration problems, and beam structural mechanics, as documented in their monographs.

The high computational efficiency, numerical stability, flexibility, and ease of implementation of the barycentric interpolation collocation method have been demonstrated. When appropriate nodes, such as Chebyshev nodes, are selected, it produces highly accurate results. Some researchers have extended the method to address 1D Burgers' equations and modified Burgers' equations [30, 31]. However, the numerical analysis and solutions for 2D coupled Burgers' equations remain relatively limited. Therefore, in this article, we aim to apply the barycentric interpolation collocation method to solve 2D coupled Burgers' equations, with the goal of further exploring its effectiveness and feasibility in numerically solving such coupled equations.

The primary objective of this paper was to develop a novel differential matrix scheme for the 2D coupled Burgers' equation. A direct linearization approach is employed to address the nonlinear terms, and a barycentric interpolation method is introduced to resolve the differentiation matrix formulation for this coupled Burgers' equation. The computational results are compared with analytical solutions across various values of the Reynolds number $R$ in selected examples. The validity and feasibility of this method are demonstrated through several illustrative cases.

The following sections are organized as follows: Section 2 presents the application of the barycentric interpolation method to develop the numerical scheme for the coupled Burgers' equation. The validity and feasibility of this method are demonstrated through several illustrative examples in Section 3. Finally, Section 4 offers conclusions and outlines potential directions for future research.

## 2. The numerical scheme for the 2D coupled Burgers' equations

In this section, we utilize the barycentric interpolation method to aid in the resolution of the differentiation matrix formulation for the 2D coupled Burgers' equations.

### 2.1. Linearization for the 2D coupled Burgers' equation

In the following, the direct linearization method is used to transform the 2D coupled Burgers' equation into a linear equation. Namely, the initial value of nonlinear term $uu_x + vu_y$ and $uv_x + vv_y$ is changed to $u^{(0)}u_x + v^{(0)}u_y$ and $u^{(0)}v_x + v^{(0)}v_y$. In this case, we have

$$\begin{cases} \frac{\partial u}{\partial t} + u^{(0)}\frac{\partial u}{\partial x} + v^{(0)}\frac{\partial u}{\partial y} - r(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) = 0, \\ \frac{\partial v}{\partial t} + u^{(0)}\frac{\partial v}{\partial x} + v^{(0)}\frac{\partial v}{\partial y} - r(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}) = 0. \end{cases} \tag{2.1}$$

By substituting $u^{(0)}$ and $v^{(0)}$ into Eq (2.1), we can calculate $u$ and $v$, denoting them as $u^{(1)}$ and $v^{(1)}$. Thus, by substituting $u^{(1)}$ and $v^{(1)}$ as the new $u^{(0)}$ and $v^{(0)}$ in Eq (2.1), we can compute new values for $u$ and $v$, which we denote as $u^{(2)}$ and $v^{(2)}$. Similarly, by substituting $u^{(s-1)}$ and $v^{(s-1)}$ in Eq (2.1), we can calculate new values for $u$ and $v$, denoted as $u^{(s)}$ and $v^{(s)}$, respectively. Thus, given the iterative representation $u \to u^{(s)}$, $u^{(0)} \to u^{(s-1)}$, $v \to v^{(s)}$, and $v^{(0)} \to v^{(s-1)}$ ($s \in N^+$ represents the number of iterations), we can construct a direct linearization scheme for Eq (2.1)

$$\begin{cases} \frac{\partial(u^{(s)})}{\partial t} + u^{(s-1)}\frac{\partial(u^{(s)})}{\partial x} + v^{(s-1)}\frac{\partial(u^{(s)})}{\partial y} - r(\frac{\partial^2(u^{(s)})}{\partial x^2} + \frac{\partial^2(u^{(s)})}{\partial y^2}) = 0, \\ \frac{\partial(v^{(s)})}{\partial t} + u^{(s-1)}\frac{\partial(v^{(s)})}{\partial x} + v^{(s-1)}\frac{\partial(v^{(s)})}{\partial y} - r(\frac{\partial^2(v^{(s)})}{\partial x^2} + \frac{\partial^2(v^{(s)})}{\partial y^2}) = 0. \end{cases} \tag{2.2}$$

Here, $u^{(s-1)}$ and $v^{(s-1)}$ denote the values of $u$ and $v$ after the $s-1$ iterations, while $\frac{\partial(u^{(s)})}{\partial t}$ and $\frac{\partial(v^{(s)})}{\partial t}$ represent the partial derivatives of $u$ and $v$ with respect to $t$ after $s$ iterations, respectively. Similarly, $\frac{\partial(u^{(s)})}{\partial x}$ and $\frac{\partial(v^{(s)})}{\partial x}$ represent the partial derivatives of $u$ and $v$ with respect to $x$ after $s$ iterations, $\frac{\partial(u^{(s)})}{\partial y}$ and $\frac{\partial(v^{(s)})}{\partial y}$ represent the partial derivatives of $u$ and $v$ with respect to $y$ after $s$ iterations, $\frac{\partial^2(u^{(s)})}{\partial x^2}$ and $\frac{\partial^2(v^{(s)})}{\partial x^2}$ represent the second-order partial derivatives of $u$ and $v$ with respect to $x$ after $s$ iterations, and $\frac{\partial^2(u^{(s)})}{\partial y^2}$, $\frac{\partial^2(v^{(s)})}{\partial y^2}$ represent the second-order partial derivatives of $u$ and $v$ with respect to $y$ after $s$ iterations, respectively.

### 2.2. *Barycentric interpolation method*

A barycentric interpolation function can be employed to deduce the approximate function of $u(x)$ for a set of $m$ distinct interpolation nodes $x_i$, $i = 1, 2, \cdots, m$, i.e.,

$$u_m(x) = \sum_{i=1}^{m} l_i(x)u(x_i) := \sum_{i=1}^{m} l_i(x)u_i, \tag{2.3}$$

and

$$l_i(x) = \frac{\omega_i}{x - x_i} / \sum_{k=1}^{m} \frac{\omega_k}{x - x_k}, \quad i = 1, 2, \cdots, m,$$

is the interpolation basis function with the interpolation weight [29, 30]

$$\omega_i = \sum_{s \in J_i} (-1)^s \prod_{g=s, g \neq i}^{s+d^*} (\frac{1}{x_i - x_g}), \quad i = 1, 2, \cdots, m, \tag{2.4}$$

and $J_i = \{s \in I_x : i - d^* \le s \le i\}$, $0 \le d^* \le m$, $I_x = \{0, 1, 2, \cdots, m\}$, so the barycentric rational interpolation is constructed. In the following discussion, the variable $d^*$ in the $x, y$, and $t$ directions is denoted as $d^*_x, d^*_y$ and $d^*_t$, respectively. For the other case

$$\omega_i = 1 / \prod_{i \neq g} (x_i - x_g), \quad i = 1, 2, \cdots, m, \tag{2.5}$$

we get the barycentric Lagrange interpolation. By observing the weights, we can notice their direct association with dispersed nodes. The weights provided are applicable to any node distribution. If commonly used node distributions such as equispaced nodes or Chebyshev nodes are selected, the computation of the barycentric weights becomes simpler. The equidistant nodes is

$$x_i = a + \frac{i - 1}{m - 1}(b - a), \quad i = 1, 2, \cdots, m,$$

and its weight function is $\omega_i = (-1)^{m-i-1}C_{m-1}^{i-1}$, and $C_{m-1}^{i-1} = \frac{(m-1)!}{(m-i)!(i-1)!}$. The Chebyshev nodes is

$$x_i = \cos\frac{(i-1)\pi}{m-1}, \quad i = 1, 2, \cdots, m,$$

and we then find that its weight function is

$$\omega_i = (-1)^i\delta_i, \quad \delta_i = \begin{cases} \frac{1}{2}, & i = 1 \; or \; m \\ 1, & ortherwise \end{cases}.$$

It is also worth noting that the Chebyshev nodes in the interval of −1 to 1 can be transformed to any interval from $a$ to $b$ through a linear transformation $\tilde{x} = \frac{b-a}{2}x + \frac{b+a}{2}$.

Solving differential equations requires the derivatives of Eq (2.3). The $p$-th derivative at $x_i$ is expressed as follows:

$$u_m^{(p)}(x) = \sum_{i=1}^{m} l_i^{(p)}(x)u_i, \quad p = 0, 1, 2, \cdots, \tag{2.6}$$

where the first-order derivative value and $p$-th-order derivative value of $l_i(x)$ at the $h$-th interpolation node are [29, 30]

$$C_h^{(1)} := l_i'(x_h) = \begin{cases} \frac{\omega_i/\omega_h}{x_h - x_i}, & h \neq i \\ -\sum\limits_{i=1, i\neq h}^{m} l_i'(x_h), & h = i \end{cases},$$

$$C_h^{(p)} := l_i^{(p)}(x_h) = \begin{cases} p(l_i^{(p-1)}(x_h)l_i'(x_h) - \frac{l_i^{(p-1)}(x_h)}{x_h - x_i}), & h \neq i, \; p \geq 2 \\ -\sum\limits_{i=1, i\neq h}^{m} l_i^{(p)}(x_h), & h = i, \; p \geq 2 \end{cases}.$$

Going a step further, if we substitute the discrete interpolation node $x_h$, $h = 1, 2, \cdots, m$ into Eq (2.3), we can obtain the discrete form as follows:

$$u^{(p)}(x_h) = \sum_{i=1}^{m} l_i^{(p)}(x_h)u_i, \quad h = 1, 2, \cdots, m. \tag{2.7}$$

Furthermore, the matrix forms of Eq (2.7) can be expressed as

$$\boldsymbol{u}^{(p)} = \boldsymbol{C}^{(p)}\boldsymbol{u}, \tag{2.8}$$

where $\boldsymbol{u} = [u_1, u_2, \cdots, u_m]^T$, $\boldsymbol{u}^{(p)} = [u_1^{(p)}, u_2^{(p)}, \cdots, u_m^{(p)}]^T$ and

$$\boldsymbol{C}^{(p)} = \begin{bmatrix} l_1^{(p)}(x_1) & l_2^{(p)}(x_1) & \cdots & l_m^{(p)}(x_1) \\ l_1^{(p)}(x_2) & l_2^{(p)}(x_2) & \cdots & l_m^{(p)}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ l_1^{(p)}(x_m) & l_2^{(p)}(x_m) & \cdots & l_m^{(p)}(x_m) \end{bmatrix}. \tag{2.9}$$

In addition, $\boldsymbol{C}^{(0)} = \boldsymbol{I}_m$, and $\boldsymbol{I}_m$ is an $m$-th-order identity matrix. The detailed derivation process of Eq (2.8) can be seen in [29].

### *2.3. Differential matrices for the coupled Burgers' equation*

In the following, we illustrate the process of establishing the differential matrix for the 2D coupled Burgers' equation. Let us consider the functions $u(x, y, t)$ and $v(x, y, t)$ defined in a spatial-temporal domain $[a, b] \times [c, d] \times (0, T]$. The spatial-temporal domain is divided by $m \times n \times l$ interpolation nodes, and the divisions of interpolation nodes in different directions are represented as

$$\begin{aligned} x &: a = x_1 < x_2 < \cdots < x_m = b; \\ y &: c = y_1 < y_2 < \cdots < y_n = d; \\ t &: 0 = t_1 < t_2 < \cdots < t_l = T. \end{aligned}$$

These interpolation nodes $(x_i, y_j, t_k)$ across the entire computing area $[a, b] \times [c, d] \times (0, T]$ can be uniform or non-uniform collocation points.

Making use of the barycentric interpolation function, we obtained the approximation function for $u(x, y, t)$ and $v(x, y, t)$, which are expressed as

$$u_{mnl}(x, y, t) = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{l} l_{ux_i}(x) l_{uy_j}(y) l_{ut_k}(t) u_{ijk}, \tag{2.10}$$

$$v_{mnl}(x, y, t) = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{l} l_{vx_i}(x) l_{vy_j}(y) l_{vt_k}(t) v_{ijk}, \tag{2.11}$$

where the symbols $u_{ijk}$ and $v_{ijk}$ represent the function values $u(x_i, y_j, t_k)$ and $v(x_i, y_j, t_k)$, respectively. Additionally, the symbols $l_{qx_i}(x)$, $l_{qy_j}(y)$, and $l_{qt_k}(t)$ $(q = u, v)$ stand for the barycentric interpolation basis functions in the directions of $x$, $y$, and $t$, respectively, and the representations of $l_{qx_i}(x)$, $l_{qy_j}(y)$, and $l_{qt_k}(t)$ $(q = u, v)$ are as follows:

$$\begin{aligned} l_{qx_i}(x) &= \frac{\omega_{x_i}}{x - x_i} \Big/ \sum_{h=1}^{m} \frac{\omega_{x_h}}{x - x_h}, \quad i = 1, 2, \cdots, m, \\ l_{qy_j}(y) &= \frac{\omega_{y_j}}{y - y_j} \Big/ \sum_{h=1}^{n} \frac{\omega_{y_h}}{y - y_h}, \quad j = 1, 2, \cdots, n, \\ l_{qt_k}(t) &= \frac{\omega_{t_k}}{t - t_k} \Big/ \sum_{h=1}^{l} \frac{\omega_{t_h}}{t - t_h}, \quad t = 1, 2, \cdots, l, \end{aligned}$$

where $\omega_{x_i}, \omega_{y_j}$, and $\omega_{t_k}$ are the corresponding weight functions.

Furthermore, we can derive the $(p_1 + p_2 + p_3)$-order partial derivative values of $u(x, y, t)$ and $v(x, y, t)$ at the interpolation nodes $\{(x_i, y_j, t_k),\ i = 1, 2, \cdots, m;\ j = 1, 2, \cdots, n;\ k = 1, 2, \cdots, l\}$ and present them as

$$u^{(p_1+p_2+p_3)}(x_i, y_j, t_k) = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{l} l_{ux_i}^{(p_1)}(x_i) l_{uy_j}^{(p_2)}(y_j) l_{ut_k}^{(p_3)}(t_k) u_{ijk}, \tag{2.12}$$

$$v^{(p_1+p_2+p_3)}(x_i, y_j, t_k) = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{l} l_{vx_i}^{(p_1)}(x_i) l_{vy_j}^{(p_2)}(y_j) l_{vt_k}^{(p_3)}(t_k) v_{ijk}. \tag{2.13}$$

Thus, Eqs (2.12) and (2.13) can be expressed in matrix form as

$$\boldsymbol{u}^{(p_1+p_2+p_3)} = \boldsymbol{C}_u^{(p_100)} \otimes \boldsymbol{C}_u^{(0p_20)} \otimes \boldsymbol{C}_u^{(00p_3)}\boldsymbol{u} := \boldsymbol{D}_u^{(p_1p_2p_3)}\boldsymbol{u}, \tag{2.14}$$

$$\boldsymbol{v}^{(p_1+p_2+p_3)} = \boldsymbol{C}_v^{(p_100)} \otimes \boldsymbol{C}_v^{(0p_20)} \otimes \boldsymbol{C}_v^{(00p_3)}\boldsymbol{v} := \boldsymbol{D}_v^{(p_1p_2p_3)}\boldsymbol{v}, \tag{2.15}$$

where $\otimes$ denotes the Kronecher product of the matrix; $\boldsymbol{C}_q^{(p_100)}$, $\boldsymbol{C}_q^{(0p_20)}$, and $\boldsymbol{C}_q^{(00p_3)}$ $(q = u, v)$ are differential matrices corresponding to $x, y$, and $t$, respectively. Their expressions are

$$\boldsymbol{C}_q^{(p_100)} = \begin{bmatrix} l_{qx_1}^{(p_1)}(x_1) & l_{qx_2}^{(p_1)}(x_1) & \cdots & l_{qx_m}^{(p_1)}(x_1) \\ l_{qx_1}^{(p_1)}(x_2) & l_{qx_2}^{(p_1)}(x_2) & \cdots & l_{qx_m}^{(p_1)}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ l_{qx_1}^{(p_1)}(x_m) & l_{qx_2}^{(p_1)}(x_m) & \cdots & l_{qx_m}^{(p_1)}(x_m) \end{bmatrix}, \tag{2.16}$$

$$\boldsymbol{C}_q^{(0p_20)} = \begin{bmatrix} l_{qy_1}^{(p_2)}(y_1) & l_{qy_2}^{(p_2)}(y_1) & \cdots & l_{qy_n}^{(p_2)}(y_1) \\ l_{qy_1}^{(p_2)}(y_2) & l_{qy_2}^{(p_2)}(y_2) & \cdots & l_{qy_n}^{(p_2)}(y_2) \\ \vdots & \vdots & \ddots & \vdots \\ l_{qy_1}^{(p_2)}(y_n) & l_{qy_2}^{(p_2)}(y_n) & \cdots & l_{qy_n}^{(p_2)}(y_n) \end{bmatrix}, \tag{2.17}$$

$$\boldsymbol{C}_q^{(00p_3)} = \begin{bmatrix} l_{qt_1}^{(p_1)}(t_1) & l_{qt_2}^{(p_1)}(t_1) & \cdots & l_{qt_l}^{(p_1)}(t_1) \\ l_{qt_1}^{(p_1)}(t_2) & l_{qt_2}^{(p_1)}(t_2) & \cdots & l_{qt_l}^{(p_1)}(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ l_{qt_1}^{(p_1)}(t_l) & l_{qt_2}^{(p_1)}(t_l) & \cdots & l_{qt_l}^{(p_1)}(t_l) \end{bmatrix}. \tag{2.18}$$

We thus obtain the matrix $\boldsymbol{C}_q^{(p_100)} \otimes \boldsymbol{C}_q^{(0p_20)} \otimes \boldsymbol{C}_q^{(00p_3)}$ of the order $(m \times n \times l) \times (m \times n \times l)$. In addition, for $x$, $\boldsymbol{C}_q^{(000)} = \boldsymbol{I}_m$; for $y$, $\boldsymbol{C}_q^{(000)} = \boldsymbol{I}_n$; for $t$, $\boldsymbol{C}_q^{(000)} = \boldsymbol{I}_l$, and $\boldsymbol{I}_m, \boldsymbol{I}_n$, and $\boldsymbol{I}_l$ are all identity matrices.

Going a step further, Eq (2.2) can be written in the format of a barycentric interpolation differential matrix

$$\begin{cases} \boldsymbol{D}_u^{(001)}\boldsymbol{u}^{(s)} + diag(\boldsymbol{u}^{(s-1)})\boldsymbol{D}_u^{(100)}\boldsymbol{u}^{(s)} + diag(\boldsymbol{v}^{(s-1)})\boldsymbol{D}_u^{(010)}\boldsymbol{u}^{(s)} - r(\boldsymbol{D}_u^{(200)} + \boldsymbol{D}_u^{(020)})\boldsymbol{u}^{(s)} = \boldsymbol{0}, \\ \boldsymbol{D}_v^{(001)}\boldsymbol{v}^{(s)} + diag(\boldsymbol{u}^{(s-1)})\boldsymbol{D}_v^{(100)}\boldsymbol{v}^{(s)} + diag(\boldsymbol{v}^{(s-1)})\boldsymbol{D}_v^{(010)}\boldsymbol{v}^{(s)} - r(\boldsymbol{D}_v^{(200)} + \boldsymbol{D}_v^{(020)})\boldsymbol{v}^{(s)} = \boldsymbol{0}, \end{cases} \tag{2.19}$$

where, $\boldsymbol{D}_q^{(001)} = \boldsymbol{I}_m \otimes \boldsymbol{I}_n \otimes \boldsymbol{C}_q^{(001)}$, $\boldsymbol{D}_q^{(100)} = \boldsymbol{C}_q^{(100)} \otimes \boldsymbol{I}_n \otimes \boldsymbol{I}_l$, $\boldsymbol{D}_q^{(010)} = \boldsymbol{I}_m \otimes \boldsymbol{C}_q^{(010)} \otimes \boldsymbol{I}_l$, $\boldsymbol{D}_q^{(200)} = \boldsymbol{C}_q^{(200)} \otimes \boldsymbol{I}_n \otimes \boldsymbol{I}_l$, $\boldsymbol{D}_q^{(020)} = \boldsymbol{I}_m \otimes \boldsymbol{C}_q^{(020)} \otimes \boldsymbol{I}_l$, $q = u, v$, and $\boldsymbol{0}$ is a zero column vector. In this case, Eq (2.19) can be expressed as a block matrix expression, i.e.,

$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{A}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{(s)} \\ \boldsymbol{v}^{(s)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \quad s = 1, 2, 3, \cdots. \tag{2.20}$$

where,

$$\boldsymbol{A}_{11} = \boldsymbol{D}_u^{(001)} + diag(\boldsymbol{u}^{(s-1)})\boldsymbol{D}_u^{(100)} + diag(\boldsymbol{v}^{(s-1)})\boldsymbol{D}_u^{(010)} - r(\boldsymbol{D}_u^{(200)} + \boldsymbol{D}_u^{(020)}),$$

$$\boldsymbol{A}_{22} = \boldsymbol{D}_v^{(001)} + diag(\boldsymbol{u}^{(s-1)})\boldsymbol{D}_v^{(100)} + diag(\boldsymbol{v}^{(s-1)})\boldsymbol{D}_v^{(010)} - r(\boldsymbol{D}_v^{(200)} + \boldsymbol{D}_v^{(020)}),$$

and $\boldsymbol{O}$ is a zero matrix.

Typically, the replacement method and the addition method are frequently utilized for the implementation of the initial-boundary conditions, as detailed in [30]. In this article, we employ the replacement method to tackle the initial and boundary conditions. After applying the boundary conditions to Eq (2.20), numerical computations can begin by setting an initial condition of $s = 1$ on the left side of Eq (2.20). As a result, the numerical solutions $\boldsymbol{u}^{(s)}$ and $\boldsymbol{v}^{(s)}$ can be computed iteratively.

## 3. Numerical experiments and discussion

In this section, we present a series of computational experiments designed to demonstrate the accuracy and effectiveness of the proposed methodology.

Within this context, we define the subsequent error standards $E^{p}_{q\infty}$ , $E^{p}_{qr}$, $(q = u, v), (p = L, R)$ to illustrate the accuracy of the proposed methodology, i.e., the error norms

$$E^{p}_{u\infty} = \|\boldsymbol{u}^{e} - \boldsymbol{u}^{c}\|_{\infty} = \max_{k} |u^{e}_{k} - u^{c}_{k}|, \ \ E^{p}_{ur} = \frac{\|\boldsymbol{u}^{e} - \boldsymbol{u}^{c}\|_{\infty}}{\|\boldsymbol{u}^{e}\|_{\infty}},$$

$$E^{p}_{v\infty} = \|\boldsymbol{v}^{e} - \boldsymbol{v}^{c}\|_{\infty} = \max_{k} |v^{e}_{k} - v^{c}_{k}|, \ \ E^{p}_{vr} = \frac{\|\boldsymbol{v}^{e} - \boldsymbol{v}^{c}\|_{\infty}}{\|\boldsymbol{v}^{e}\|_{\infty}}.$$

Among these, $\boldsymbol{q}^{c}$ and $\boldsymbol{q}^{e}$ denotes the numerical and analytical solutions, respectively, and $\boldsymbol{q} = \boldsymbol{u}, \boldsymbol{v}$. Specifically, the parameter $p$ is defined as $L$ in relation to barycentric Lagrange interpolation and as $R$ in relation to barycentric rational interpolation. In all subsequent tables, we denote the absolute error under equidistant nodes and Chebyshev nodes as $E^{pd}_{q\infty}$ and $E^{pc}_{q\infty}$, respectively, where $q = u, v$ and $p = L, R$. Similarly, the relative error under equidistant nodes and Chebyshev nodes is represented as $E^{pd}_{qr}$ and $E^{pc}_{qr}$, respectively, with the same definitions for $q$ and $p$.

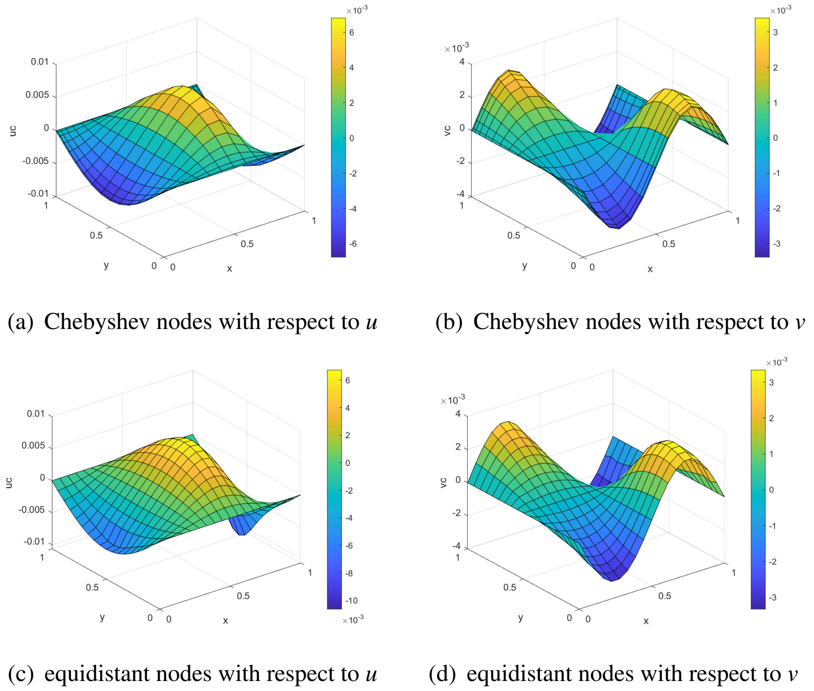
**Example 1.** *Eq* (1.1) *with exact solutions*

$$u(x, y, t) = -2r\frac{2\pi\cos(2\pi x)\sin(\pi y)\exp(-5\pi^{2}rt)}{\sin(2\pi x)\sin(\pi y)\exp(-5\pi^{2}rt) + 2},$$

$$v(x, y, t) = -2r\frac{\pi\sin(2\pi x)\cos(\pi y)\exp(-5\pi^{2}rt)}{\sin(2\pi x)\sin(\pi y)\exp(-5\pi^{2}rt) + 2},$$

$(x, y) \in [0, 1] \times [0, 1]$, $t \geq 0$ *is under consideration in this example, and its initial–boundary conditions can be established on the basis of the exact solution.*

In Example 1, the numerical solutions obtained from the Chebyshev and equidistant node calculations are illustrated in Figures 1 and 2. Furthermore, we present the error distribution calculated by barycentric interpolation, with the parameters set to $m = n = l = 18$, $R = 1000$, $t = 1$, and $d^{*}_{x} = d^{*}_{y} = d^{*}_{t} = 6$ in Figures 3 and 4. The variations in the absolute and relative errors associated with equidistant nodes and Chebyshev nodes, determined using different quantities of nodes, are summarized in Tables 1–4. Additionally, Table 5 offers a comprehensive overview of the errors calculated for Chebyshev nodes under varying values of $R$ with $m = n = l = 18$, $d^{*}_{x} = d^{*}_{y} = d^{*}_{t} = 6$, and $t = 1$. This example illustrates that our algorithm maintains a high level of computational accuracy, even for larger values of $R$.



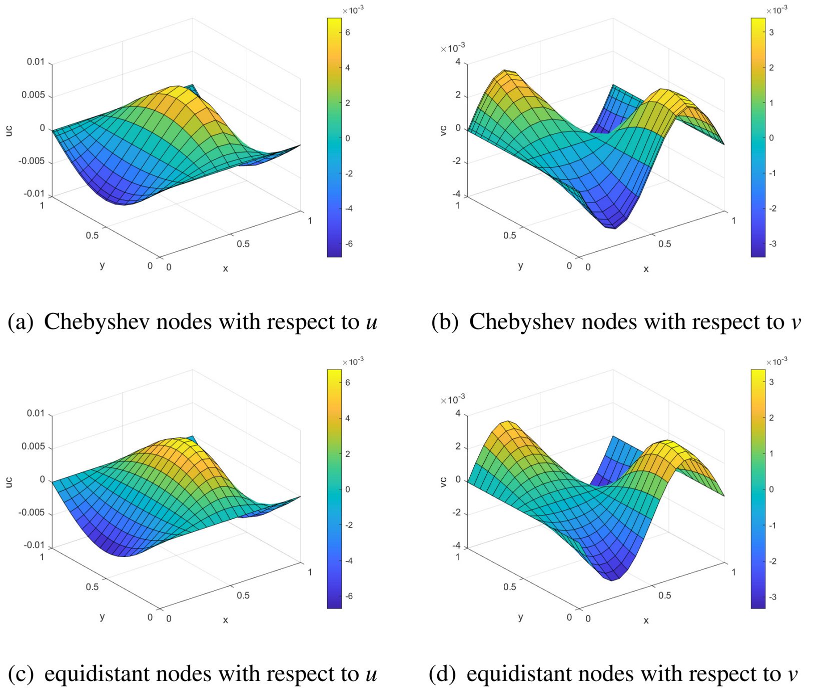
(a) Chebyshev nodes with respect to $u$

(b) Chebyshev nodes with respect to $v$

(c) equidistant nodes with respect to $u$

(d) equidistant nodes with respect to $v$

**Figure 1.** The numerical solutions calculated by the barycentric Lagrange interpolation with $m = n = l = 18$ , $R = 1000$, and $t = 1$ for Example 1.



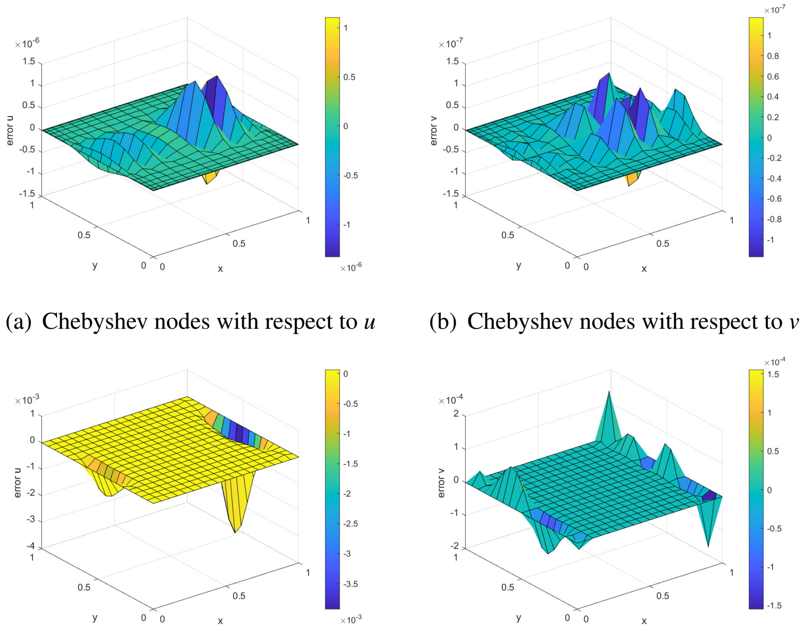
(a) Chebyshev nodes with respect to $u$

(b) Chebyshev nodes with respect to $v$

(c) equidistant nodes with respect to $u$

(d) equidistant nodes with respect to $v$

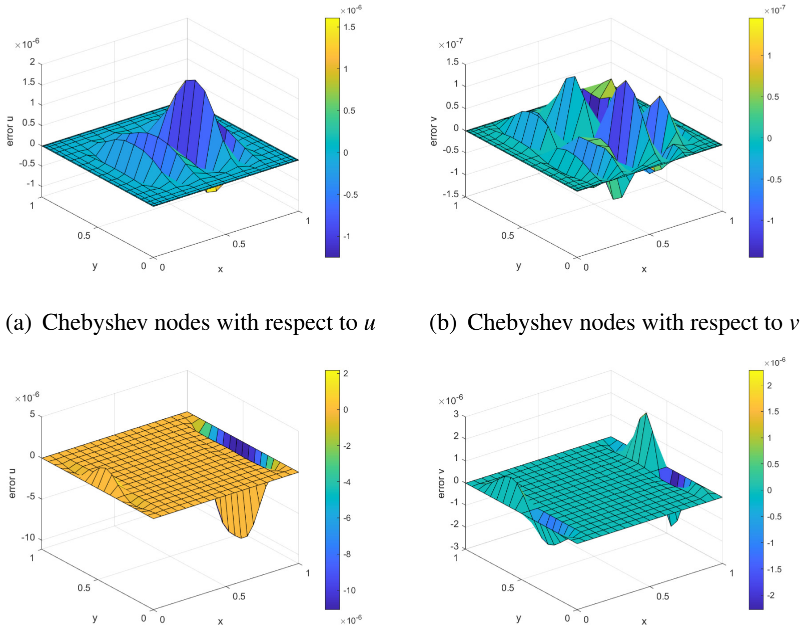
**Figure 2.** The numerical solutions calculated by the barycentric rational interpolation with $m = n = l = 18$, $R = 1000$, $t = 1$, and $d_x^* = d_y^* = d_t^* = 6$ for Example 1.



(a) Chebyshev nodes with respect to $u$ (b) Chebyshev nodes with respect to $v$

(c) equidistant nodes with respect to $u$ (d) equidistant nodes with respect to $v$

**Figure 3.** The error distribution calculated by the barycentric Lagrange interpolation with $m = n = l = 18$ , $R = 1000$, and $t = 1$ for Example 1.



(a) Chebyshev nodes with respect to $u$ (b) Chebyshev nodes with respect to $v$

(c) equidistant nodes with respect to $u$ (d) equidistant nodes with respect to $v$

**Figure 4.** The error distribution calculated by the barycentric rational interpolation with $m = n = l = 18$ , $R = 1000$, and $t = 1$ for Example 1.

**Table 1.** Errors of $u$ calculated using the barycentric Lagrange collocation methods with $R = 1000$ for Example 1.

| $m, n, l$ | $E^{Lc}_{u\infty}$ | $E^{Lc}_{ur}$ | $E^{Ld}_{u\infty}$ | $E^{Ld}_{ur}$ |
|---|---|---|---|---|
| 3 | $4.5802 \times 10^{-5}$ | $1.0831 \times 10^{-2}$ | $1.9745 \times 10^{-4}$ | $3.8122 \times 10^{-2}$ |
| 6 | $1.2615 \times 10^{-4}$ | $1.8616 \times 10^{-2}$ | $6.6265 \times 10^{-4}$ | $1.1080 \times 10^{-1}$ |
| 9 | $3.1096 \times 10^{-5}$ | $4.8138 \times 10^{-3}$ | $9.2192 \times 10^{-4}$ | $1.3987 \times 10^{-1}$ |
| 12 | $1.4204 \times 10^{-5}$ | $2.0962 \times 10^{-3}$ | $1.6105 \times 10^{-3}$ | $2.3695 \times 10^{-1}$ |
| 15 | $4.3054 \times 10^{-6}$ | $6.4973 \times 10^{-4}$ | $2.2737 \times 10^{-3}$ | $3.3789 \times 10^{-1}$ |

**Table 2.** Errors of $v$ calculated using the barycentric Lagrange collocation methods with $R = 1000$ for Example 1.

| $m, n, l$ | $E^{Lc}_{v\infty}$ | $E^{Lc}_{vr}$ | $E^{Ld}_{v\infty}$ | $E^{Ld}_{vr}$ |
|---|---|---|---|---|
| 3 | $1.3031 \times 10^{-4}$ | $4.0889 \times 10^{-2}$ | $7.9112 \times 10^{-5}$ | $3.0549 \times 10^{-2}$ |
| 6 | $3.2023 \times 10^{-5}$ | $9.8617 \times 10^{-3}$ | $5.3732 \times 10^{-5}$ | $1.9021 \times 10^{-2}$ |
| 9 | $1.0720 \times 10^{-5}$ | $3.1844 \times 10^{-3}$ | $5.2997 \times 10^{-5}$ | $1.6081 \times 10^{-2}$ |
| 12 | $2.7465 \times 10^{-6}$ | $8.0844 \times 10^{-4}$ | $6.3429 \times 10^{-5}$ | $1.8664 \times 10^{-2}$ |
| 15 | $5.5838 \times 10^{-7}$ | $1.6428 \times 10^{-4}$ | $8.1760 \times 10^{-5}$ | $2.4300 \times 10^{-2}$ |

**Table 3.** Errors of $u$ calculated using the barycentric rational collocation methods with $R = 1000$ for Example 1.

| $m, n, l$ | $E^{Rc}_{u\infty}$ | $C_o$ | $E^{Rc}_{ur}$ | $E^{Rd}_{u\infty}$ | $E^{Rd}_{ur}$ |
|---|---|---|---|---|---|
| 6 | $1.2615 \times 10^{-4}$ | - | $1.8616 \times 10^{-2}$ | $6.6265 \times 10^{-4}$ | $1.1080 \times 10^{-1}$ |
| 9 | $3.1330 \times 10^{-5}$ | 3.44 | $4.8500 \times 10^{-3}$ | $5.0975 \times 10^{-5}$ | $7.7339 \times 10^{-2}$ |
| 12 | $1.6498 \times 10^{-5}$ | 2.23 | $2.4347 \times 10^{-3}$ | $5.8910 \times 10^{-5}$ | $8.6673 \times 10^{-3}$ |
| 15 | $9.6679 \times 10^{-6}$ | 2.40 | $1.4590 \times 10^{-3}$ | $8.8309 \times 10^{-5}$ | $1.3123 \times 10^{-2}$ |
| 18 | $1.6033 \times 10^{-6}$ | 9.85 | $2.3616 \times 10^{-4}$ | $1.1085 \times 10^{-5}$ | $1.6514 \times 10^{-3}$ |

**Table 4.** Errors of $v$ calculated using the barycentric rational collocation methods with $R = 1000$ for Example 1.

| $m, n, l$ | $E^{Rc}_{v\infty}$ | $E^{Rc}_{vr}$ | Time | $E^{Rd}_{v\infty}$ | $E^{Rd}_{vr}$ | Time |
|---|---|---|---|---|---|---|
| 6 | $3.2023 \times 10^{-5}$ | $9.8617 \times 10^{-3}$ | 0.20s | $5.3732 \times 10^{-5}$ | $1.9021 \times 10^{-2}$ | 0.27s |
| 9 | $1.0595 \times 10^{-5}$ | $3.1473 \times 10^{-3}$ | 0.84s | $1.8742 \times 10^{-5}$ | $5.6869 \times 10^{-3}$ | 1.02s |
| 12 | $6.8583 \times 10^{-6}$ | $2.0187 \times 10^{-3}$ | 4.47s | $3.9817 \times 10^{-5}$ | $1.1716 \times 10^{-2}$ | 4.68s |
| 15 | $6.5547 \times 10^{-7}$ | $1.9285 \times 10^{-4}$ | 18.60s | $2.8833 \times 10^{-6}$ | $8.5696 \times 10^{-4}$ | 15.79s |
| 18 | $1.4502 \times 10^{-7}$ | $4.2773 \times 10^{-5}$ | 87.00s | $2.2777 \times 10^{-6}$ | $6.8380 \times 10^{-4}$ | 58.78s |

**Table 5.** Errors of $u$, $v$ calculated using the barycentric rational collocation methods at $t = 1$ with $m = n = l = 18$, $d_x^* = d_y^* = d_t^* = 6$, and different valsues of $R$ for Example 1.

| $R$ | $E_{u\infty}^{Lc}$ | $E_{v\infty}^{Lc}$ | $E_{u\infty}^{Rc}$ | $E_{v\infty}^{Rc}$ |
|---|---|---|---|---|
| 10 | $6.4443 \times 10^{-8}$ | $3.2138 \times 10^{-9}$ | $4.5280 \times 10^{-8}$ | $1.4181 \times 10^{-8}$ |
| 50 | $2.2821 \times 10^{-7}$ | $1.8697 \times 10^{-8}$ | $1.9123 \times 10^{-6}$ | $3.3280 \times 10^{-7}$ |
| 100 | $1.2155 \times 10^{-6}$ | $6.0399 \times 10^{-8}$ | $7.2169 \times 10^{-6}$ | $8.2106 \times 10^{-7}$ |
| 1000 | $1.3332 \times 10^{-6}$ | $1.1720 \times 10^{-7}$ | $1.6033 \times 10^{-6}$ | $1.4502 \times 10^{-7}$ |
| 2000 | $2.9127 \times 10^{-6}$ | $1.7987 \times 10^{-7}$ | $4.7928 \times 10^{-7}$ | $4.5517 \times 10^{-8}$ |
| 5000 | $9.8039 \times 10^{-8}$ | $1.7777 \times 10^{-8}$ | $8.4328 \times 10^{-8}$ | $8.3866 \times 10^{-9}$ |
| 10000 | $2.6571 \times 10^{-8}$ | $8.1309 \times 10^{-9}$ | $2.1664 \times 10^{-8}$ | $2.2004 \times 10^{-9}$ |

Additionally, to quantitatively assess the convergence order ($C_o$) of the numerical scheme, the formula $C_o = \log \frac{E_i}{E_{i+1}} / \log \frac{h_i}{h_{i+1}}$ is applied to analyze the data presented in Table 3, where $E_i$ represents the numerical error for the $i$-th mesh resolution, and $h_i$ denotes the maximum stride length of the partition corresponding to the variables in the $i$-th mesh. The results indicate that the calculation of this method is convergent. The CPU times are also given in Table 4, which represent the computational efficiency of this method, and the results indicate that the computational cost of this method is lower than that of other methods.

**Example 2.** *We consider Eq* (1.1) *with exact conditions*

$$u(x, y, t) = 0.25[3 - (1 + \exp((-4x + 4y - t)/(32r)))^{-1}],$$

$$v(x, y, t) = 0.25[3 + (1 + \exp((-4x + 4y - t)/(32r)))^{-1}],$$

$(x, y) \in [0, 1] \times [0, 1]$ *and* $t \geq 0$. *Furthermore, the initial and boundary conditions are delineated in accordance with the exact solutions.*

The error distribution associated with the barycentric Lagrange collocation method, calculated using Chebyshev nodes with the parameters $m = n = l = 18$ and varying values of $R$, is illustrated in Figures 5–7 for Example 2. The numerical solutions obtained for $R = 10, 20, 50, 200$ demonstrate a high degree of accuracy, making it challenging to differentiate between the exact solutions and the numerical approximations in the figures, particularly in Figures 5 and 6. In this context, the proposed algorithm exhibits effective performance for smaller values of $R$, as indicated by the numerical results. However, its effectiveness appears to diminish for larger values of $R$. It is plausible that despite the initial data being smooth, a solution shock may develop as time progresses. In the event of a solution shock, it is possible to adjust the number of computational nodes to manage the shock and enhance the accuracy of the algorithm. To further illustrate the accuracy, the numerical and exact solutions for $R = 500$ at $t = 1$ , with $(x, y) \in [0, 0.5] \times [0, 0.5]$, are presented in Figure 8.

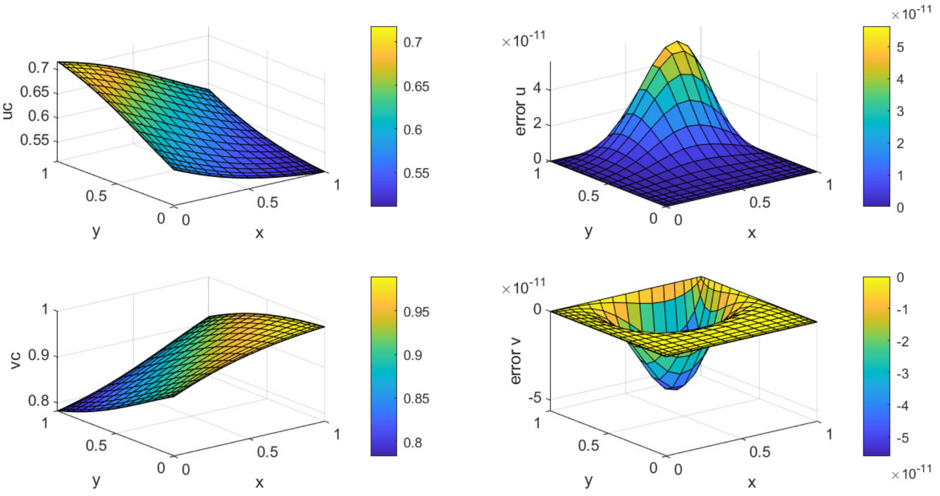


**Figure 5.** Comparison between the numerical and exact solutions at $t = 1$ with $R = 20$ and $D : [0, 1] \times [0, 1]$ for Example 2.

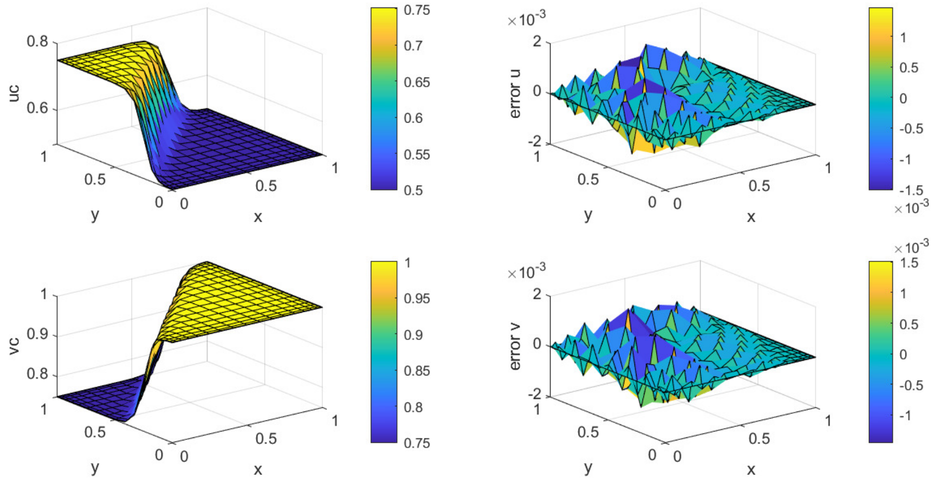


**Figure 6.** Comparison between the numerical and exact solutions at $t = 1$ with $R = 200$ and $D : [0, 1] \times [0, 1]$ for Example 2.

**Example 3.** *Here, the 2D Burgers' equation [14]*

$$v_t + v(v_x + v_y) = r(v_{xx} + v_{yy}) \tag{3.1}$$

*is considered with the exact condition*

$$v(x, y, t) = [1 + \exp((x + y - t)/(2r))]^{-1}, \quad (x, y) \in [0, 2] \times [0, 2], \quad t \geq 0.$$

Similarly, the initial condition and boundary conditions are all specified by the given exact solution. We can bring in a complementary function $u(x, y, t) = v(x, y, t)$, which can transform Eq (3.1) with only a single equation and a single unknown function $v(x, y, t)$ into a coupled Burgers' equation, analogous to Eq (1.1), and then apply the method proposed in this paper. The corresponding differential matrices of Burgers' equation (3.1) can be expressed as

$$[\boldsymbol{D}_v^{(001)} + diag(\boldsymbol{v}^{(s-1)})(\boldsymbol{D}_v^{(100)} + \boldsymbol{D}_v^{(010)}) - r(\boldsymbol{D}_v^{(200)} + \boldsymbol{D}_v^{(020)})]\boldsymbol{v}^{(s)} = \boldsymbol{0}, \ \ s = 1, 2, 3, \cdots . \tag{3.2}$$

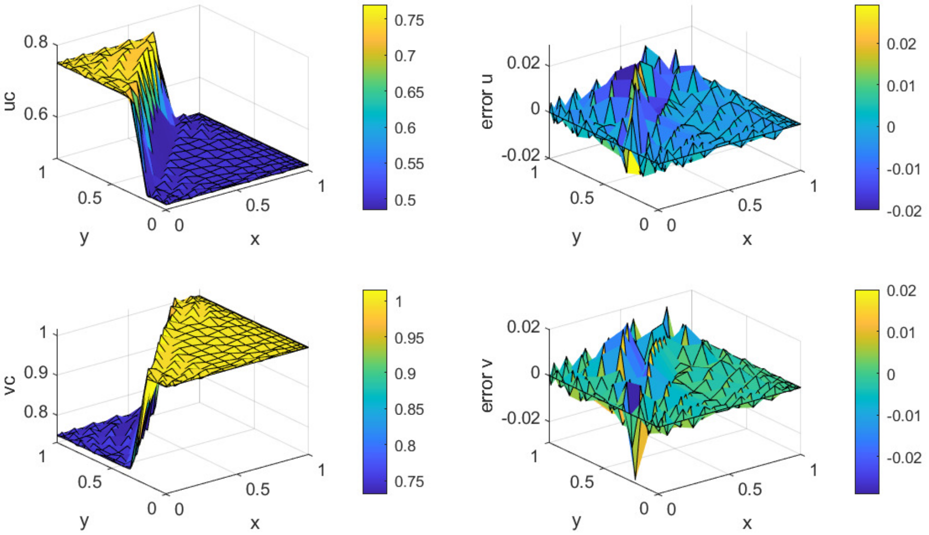


**Figure 7.** Comparison between the numerical and exact solutions at $t = 1$ with $R = 500$ and $D : [0, 1] \times [0, 1]$ for Example 2.

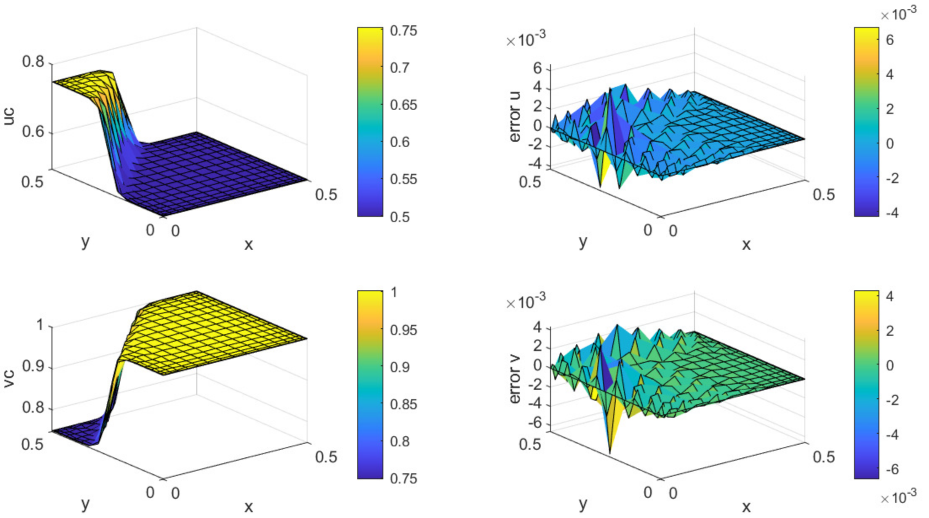
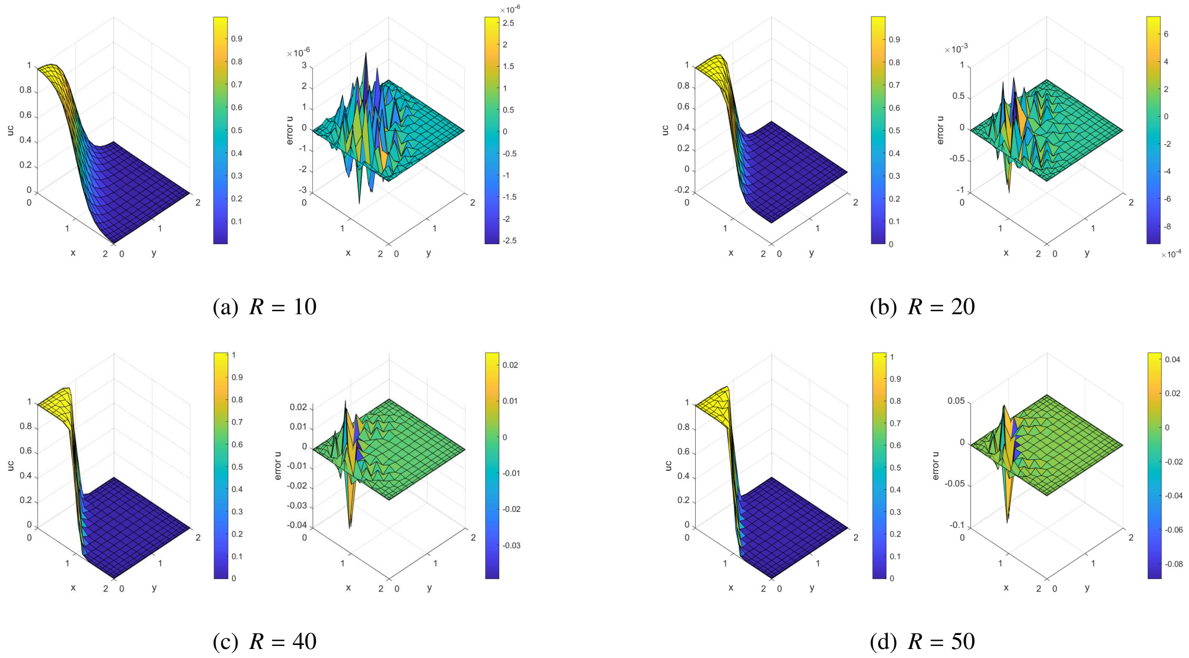


**Figure 8.** Comparison between the numerical and exact solutions at $t = 1$ with $R = 500$ and $D : [0, 0.5] \times [0, 0.5]$ for Example 2.



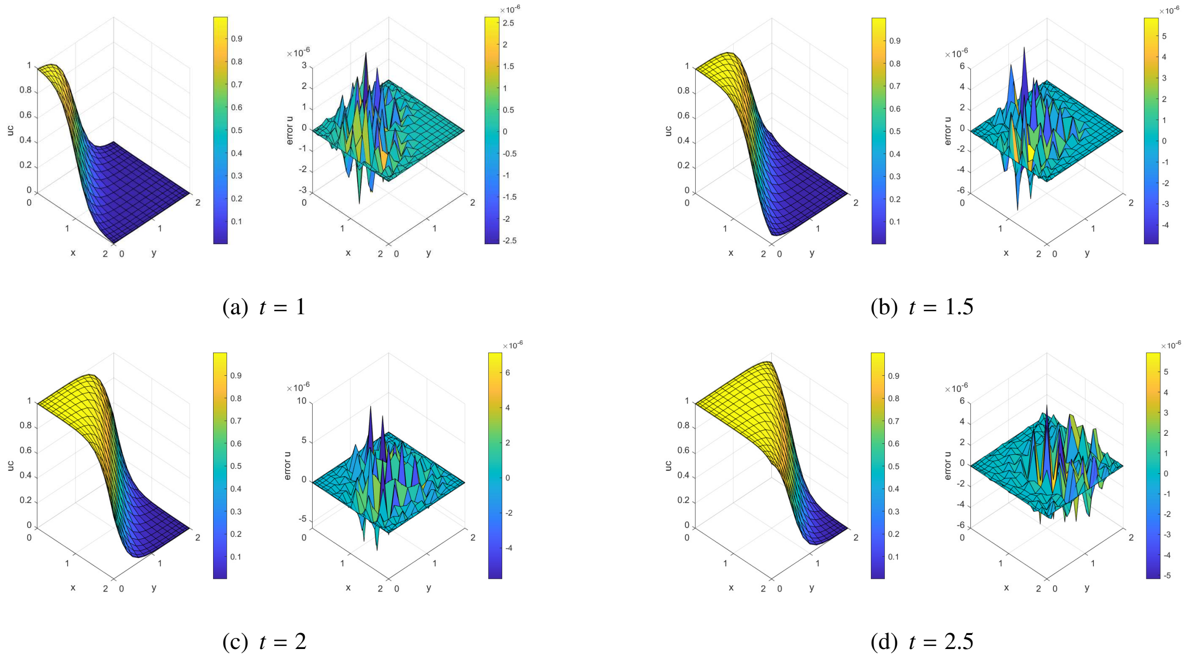
(a) $R = 10$

(b) $R = 20$

(c) $R = 40$

(d) $R = 50$

**Figure 9.** Comparison between the numerical and exact solutions at $t = 1$ with different $R$ and $D : [0, 2] \times [0, 2]$ for Example 3.



(a) $t = 1$

(b) $t = 1.5$

(c) $t = 2$

(d) $t = 2.5$

**Figure 10.** Comparison between the numerical and exact solutions at different $t$ with $R = 10$ and $D : [0, 2] \times [0, 2]$ for Example 3.

The numerical results obtained through the barycentric rational interpolation method, utilizing various values of $R$, are presented in Figure 9. As $R$ increases, the solutions exhibit steeper gradients. For example, when $R = 40, 50$, the numerical solutions illustrated in Figure 9(c),(d) exhibit steeper gradients compared with those shown in Figure 9(a),(b). Additionally, the region where the error exceeds $10^{-3}$ is confined to a small area near the line defined by $x + y = t$. To further elucidate the long-term behavior of this scheme, we present the numerical solutions and the corresponding errors for various values of $t$ in Figure 10. It is apparent that the region characterized by a steep slope shifts as $t$ increases, reflecting the behavior of the exact solutions depicted in Figure 11.

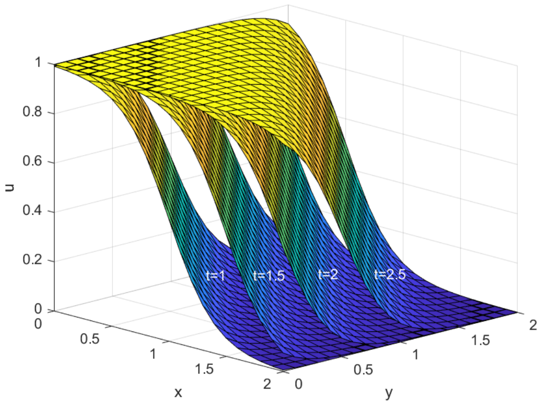
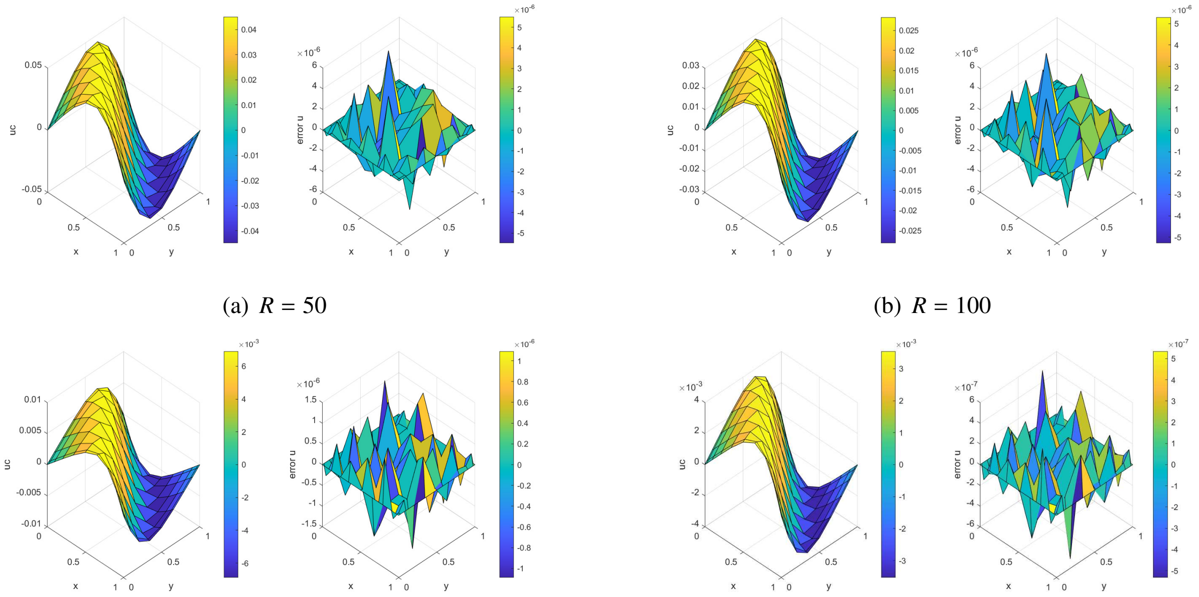


**Figure 11.** The exact solutions at $t = 1, 1.5, 2, 2.5$ with $R = 10$ and $D : [0, 2] \times [0, 2]$ for Example 3.



(a) $R = 50$

(b) $R = 100$

(c) $R = 500$

(d) $R = 1000$

**Figure 12.** The error distribution calculated by the barycentric rational interpolation at $t = 1$ with $m = n = l = 10$, $d_x^* = d_y^* = d_t^* = 6$ for Example 4.

**Example 4.** *We consider the Eq* (3.1) *with the exact solution [32]*

$$v(x, y, t) = \frac{2r\pi \sin(\pi x + \pi y) \exp(-2r\pi^2 t)}{2 + \cos(\pi x + \pi y) \exp(-2r\pi^2 t)}$$

*over a domain* $(x, y) \in [0, 1] \times [0, 1]$ *and* $t \geq 0$*, with the initial and boundary conditions specified by the given exact solution.*

In contrast to Example 3, the present example demonstrates that even when the value of $R$ is relatively large and the number of computing nodes is configured to $m = n = l = 8$, the computational accuracy of this method remains consistently stable, achieving an accuracy level of up to $10 \times 10^{-6}$. To facilitate a clearer understanding of the differences between the numerical and exact solutions, we present both the exact and numerical solutions for $R = 50, 100, 500, 1000$ at $t = 1$ in Figure 12. It is evident that, despite the relatively large value of $R$, the solutions remain smooth and devoid of shocks. The minimal discrepancies observed between the exact and numerical solutions further underscore the high accuracy of the method.

## 4. Conclusions

This paper presents an efficient methodology for the numerical solution of a 2D coupled Burgers' equation. Specifically, the barycentric interpolation collocation method is employed to approximate the unknown function, while a specially constructed differential matrix is utilized to derive the numerical solution of the 2D coupled Burgers' equation. The results from numerical experiments demonstrate the effectiveness and accuracy of the proposed scheme in solving various instances of Burgers' equations. Furthermore, it is shown that the current algorithm can yield satisfactory numerical results for larger parameter values of $R$, specifically for $R \geq 1000$. It is important to acknowledge that the method proposed in this article has certain limitations and challenges when calculating the coupled equations in irregular regions. In the future, we will continue to explore the application of this method for solving the equations defined in these irregular regions and will conduct further theoretical analyses of coupled equations, including convergence analysis. Additionally, we plan to extend this method to tackle equations that have not been investigated in other fields, as well as coupled equations of other different categories.

## 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 was supported by the Natural Science Foundation of Shandong Province (Grant No. ZR2022MA003).

## Conflict of interest

The authors declare there is no conflicts of interest.

## References

1. M. P. Bonkile, A. Awasthi, C. Lakshmi, V. Mukundan, V. S. Aswin, A systematic literature review of Burgers' equation with recent advances, *Pramana*, **90** (2018), 69. https://doi.org/10.1007/s12043-018-1559-4

2. M. Khan, A novel solution technique for two dimensional Burger's equation, *Alexandria Eng. J.*, **53** (2014), 485–490. https://doi.org/10.1016/j.aej.2014.01.004

3. R. Jiwari, R. C. Mittal, K. K. Sharma, A numerical scheme based on weighted average differential quadrature method for the numerical solution of Burgers' equation, *Appl. Math. Comput.*, **219** (2013), 6680–6691. https://doi.org/10.1016/j.amc.2012.12.035

4. M. Kumar, S. Pandit, A composite numerical scheme for the numerical simulation of coupled Burgers' equation, *Comput. Phys. Commun.*, **185** (2014), 809–817. https://doi.org/10.1016/j.cpc.2013.11.012

5. M. Tamsir, V. K. Srivastava, R. Jiwari, An algorithm based on exponential modified cubic B-spline differential quadrature method for nonlinear Burgers' equation, *Appl. Math. Comput.*, **290** (2016), 111–124. https://doi.org/10.1016/j.amc.2016.05.048

6. S. Bak, P. Kim, D. Kim, A semi-Lagrangian approach for numerical simulation of coupled Burgers' equations, *Commun. Nonlinear Sci. Numer. Simul.*, **69** (2019), 31–44. https://doi.org/10.1016/j.cnsns.2018.09.007

7. 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

8. H. Aminikhah, An analytical approximation for coupled viscous Burgers' equation, *Appl. Math. Modell.*, **37** (2013), 5979–5983. https://doi.org/10.1016/j.apm.2012.12.013

9. H. Eltayeb, I. Bachar, A note on singular two-dimensional fractional coupled Burgers' equation and triple Laplace Adomian decomposition method, *Bound. Value Probl.* , **2020** (2020), 129. https://doi.org/10.1186/s13661-020-01426-0

10. R. K. Alhefthi, H. Eltayeb, The Solution of two-dimensional coupled Burgers' equation by G-double laplace transform, *J. Funct. Spaces*, **2023** (2023), 4320612. https://doi.org/10.1155/2023/4320612

11. D. Kaya, An explicit solution of coupled viscous Burgers' equation by the decomposition method, *Int. J. Math. Math. Sci.*, **27** (2001), 675–680. https://doi.org/10.1155/S0161171201010249

12. J. Biazar, H. Ghazvini, Exact solutions for nonlinear Burgers' equation by homotopy perturbation method, *Numer. Methods Partial Differ. Equations*, **25** (2009), 833–842. https://doi.org/10.1002/num.20376

13. 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

14. 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

15. E. Hopf, The partial differential equation $u_t + uu_x = u_{xx}$, *Commun. Pure Appl. Math.*, **3** (1950), 201–230. https://doi.org/10.1002/cpa.3160030302

16. J. D. Cole, On a quasilinear parabolic equations occurring in aerodynamics, *Q. Appl. Math.*, **9** (1951), 225–236. https://doi.org/10.1090/qam/42889

17. C. A. J. Fletcher, Generating exact solutions of the two-dimensional Burgers' equations, *Int. J. Numer. Methods Fluids*, **3** (1983), 213–216. https://doi.org/10.1002/fld.1650030302

18. Y. Chai, J. Ouyang, Appropriate stabilized Galerkin approaches for solving two-dimensional coupled Burgers' equations at high Reynolds numbers, *Comput. Math. Appl.* , **79** (2020), 1287–1301. https://doi.org/10.1016/j.camwa.2019.08.036

19. A. J. Hussein, H. A. Kashkool, A weak Galerkin finite element method for two-dimensional coupled Burgers' equation by using polynomials of order $(k, k-1, k-1)$, *J. Interdiscip. Math.*, **23** (2020), 777–790. https://doi.org/10.1080/09720502.2019.1706844

20. 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

21. Y. Uçar, N. M. Yağmurlu, M. K. Yiğit, Numerical Solution of the coupled Burgers equation by trigonometric B-spline collocation method, *Math. Methods Appl. Sci.*, **46** (2023), 6025–6041. https://doi.org/10.1002/mma.8887

22. Z. Dehghan, J. Rashidinia, Numerical solution of coupled viscous Burgers' equations using RBF-QR method, *Math. Sci.*, **17** (2023), 317–324. https://doi.org/10.1007/s40096-022-00472-2

23. S. Park, Y. Jeon, P. Kim, S. Bak, An error predict-correction formula of the load vector in the BSLM for solving three-dimensional Burgers' equations, *Math. Comput. Simul.*, **221** (2024), 222–243. https://doi.org/10.1016/j.matcom.2024.03.001

24. S. Torkaman, M. Heydari, G. B. Loghmani, Piecewise barycentric interpolating functions for the numerical solution of Volterra integro-differential equations, *Math. Methods Appl. Sci.* , **45** (2022), 6030–6061. https://doi.org/10.1002/mma.8154

25. H. Liu, J. Huang, Y. Pan, J. Zhang, Barycentric interpolation collocation method for solving linear and nonlinear high-dimensional Fredholm integral equations, *J. Comput. Appl. Math.*, **327** (2018), 141–154. https://doi.org/10.1016/j.cam.2017.06.004

26. J. Li, Y. Cheng, Linear barycentric rational collocation method for solving heat conduction equation, *Numer. Methods Partial Differ. Equations*, **37** (2021), 533–545. https://doi.org/10.1002/num.22539

27. J. Li, Linear barycentric rational collocation method to solve plane elasticity problems, *Math. Biosci. Eng.*, **20** (2023), 8337–8357. https://doi.org/10.3934/mbe.2023365

28. J. Li, Barycentric rational collocation method for semi-infinite domain problems, *AIMS Math.*, **8** (2023), 8756–8771. https://doi.org/10.3934/math.2023439

29. S. Li, Z. Wang, *Meshless Barycentric Interpolation Collocation Method-Algorithmics, Programs & Applications in Engineering*, Science Publishing, Beijing, 2012.

30. Z. Wang, S. Li, *Barycentric Interpolation Collocation Method for Nonlinear Problems*, National Defense Industry Press, Beijing, 2015.

31. Ö. Oruç, Two meshless methods based on pseudo spectral delta-shaped basis functions and barycentric rational interpolation for numerical solution of modified Burgers equation, *Int. J. Comput. Math.*, **98** (2021), 461–479. https://doi.org/10.1080/00207160.2020.1755432

32. Y. Hu, A. Peng, L. Chen, Y. Tong, Z. Weng, A finite difference-collocation method for the Burgers equation, *Pure Appl. Math.*, **39** (2023), 100–112.