Int. J. of Applied Mechanics and Engineering, 2022, vol.27, No.3, pp.79-102
DOI: 10.2478/ijame-2022-0037

A NUMERICAL APPROXIMATION OF 2D COUPLED BURGERS' EQUATION USING MODIFIED CUBIC TRIGONOMETRIC B-SPLINE DIFFERENTIAL QUADRATURE METHOD

Mamta KAPOOR* and Varun JOSHI
Department of Mathematics, Lovely Professional University, Punjab,144411, INDIA
E-mail: mamtakapoor.78@yahoo.com

In the present paper, trigonometric B-spline DQM is applied to get the approximated solution of coupled 2D non-linear Burgers' equation. This technique, named modified cubic trigonometric B-spline DQM, has been used to obtain accurate and effective numerical approximations of the above-mentioned partial differential equation. For checking the compatibility of results, different types of test examples are discussed. A comparison is done between L_2 and L_∞ error norms with the previous, present results and with the exact solution. The resultant set of ODEs has been solved by employing the SSP RK 43 method. It is observed that the obtained results are improved compared to the previous numerical results in the literature.

Key words: trigonometric B-spline, differential quadrature method (DQM), strong stability preserving Runge-Kutta 43 (SSP-RK43) scheme, L_2 and L_∞ error norms; coupled 2D non-linear Burgers' equation.

1. Introduction

In this paper, the considered equation is 2D non-linear coupled Burgers' equation as follows:

$$\frac{\partial u}{\partial t}+u\ \frac{\partial u}{\partial x}+v\ \frac{\partial v}{\partial y}=\left(\frac{1}{Re}\right)\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right), \tag{1.1}$$

$$\frac{\partial v}{\partial t}+u\ \frac{\partial v}{\partial x}+v\ \frac{\partial v}{\partial y}=\left(\frac{1}{Re}\right)\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right). \tag{1.2}$$

Initial conditions:

$$\left.\begin{aligned} u(x,\ y,0)=\Phi_1(x,y) \\ v(x,\ y,0)=\Phi_2(x,y) \end{aligned}\right\}, \qquad (x,y)\in D\ . \tag{1.3}$$

Where, D is the computational domain which is square given as follows:

$$D=\left\{(x,y): a\le x\le b, c\le y\le d\right\}. \tag{1.4}$$

* To whom correspondence should be addressed

Boundary conditions:

$$\left.\begin{aligned} u(x, y, t) &= \psi_1(x, y, t) \\ v(x, y, t) &= \psi_2(x, y, t) \end{aligned}\right\}, \quad (x,y)\in \partial D \quad \text{and} \quad t>0. \tag{1.5}$$

Where, ∂D is the boundary of a given computational domain, $u(x,t)$ is the component of velocity in one dimension, $u(x,y,t)$ and $v(x,y,t)$ are the components of velocity in $2D$. Φ_1,Φ_2, ψ_1,ψ_2 are the known functions, $\frac{\partial u}{\partial t}$ is the unsteady term, $u\,\frac{\partial u}{\partial x}$ is the non-linear convection term, $\upsilon\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)$ is known as a diffusion term, υ is the viscosity coefficient greater than 0, Re is known as the Reynolds number. Coupled viscous Burgers' equation is the most suitable arrangement of the Navier-Stokes equation, which also has exact solutions. In the coupled viscous Burgers' equation convection and diffusion are taken into consideration and the equation is equivalent to the incompressible Navier-Stokes equation. In order to develop the methods for computation of complex fluids, the first step is to attain the numerical solution of Burgers' equation. That is why developing new schemes to get the numerical approximations of the Burgers' equation is a novelty.

In 1983 Fletcher [1] gave analytical solutions of the 2D Burgers' equation by using the Hopf-Cole transformations. In previous years much effort has been made in order to get the numerical approximation of the same equation by many researchers [2-12]. In the last decade, a lot of work has been done upon the Burgers' equation. The Burgers' equation was unraveled by implementing various new schemes, in order to get analytical as well as numerical solutions like in [1, 13]. The equation was solved by Hopf-Cole transformations, by implementing finite difference methods [14-18], by making use of B-spline collocation method [19], by using polynomial DQM [20-21], by implementing quartic the B-spline DQM [22], by using the modified cubic B-spline based collocation method [23], by the cubic B-spline based DQM [24], by implementing the scheme of modified cubic B-spline based DQM [25, 26] and many others. Mittal and Jiwari [28] used DQM to solve the 2D coupled Burgers' equation, and its convergence and stability were also discussed. Jain and Holla [2] implemented the cubic B-spline regime to solve 2D coupled Burgers' equation. They also discussed stability and convergence. Zhu *et al.* [29] anticipated the discrete Adomian Decomposition Method in order to obtain the numerical approximation of the 2D coupled Burgers' equation. Hossein [30] proposed a scheme of new hybrid Laplace and New Homotopy Perturbation Method (LTNHPM) to solve the 2D coupled Burgers' equation.

The differential quadrature method is a numerical regime for obtaining the solution of a range of partial differential equations. Basically, DQM was established by Bellman *et al.* in the 1970s. DQM is similar to the integral quadrature method, and with the help of DQM, an approximation of derivative of a function at any point can be made by employing the linear sum of the complete set of functional values along a given mesh line. The main idea of DQM lies in obtaining the weighting coefficients. DQM and its different applications got developed by leaps and bounds in the late 1980s. In 1996 a review of sequential development and applications of DQM was given by Bert and Malik [32]. Chang Shu [33] presented a simple algebraic formulation to obtain the weighting coefficients for the approximation of the first-order derivative with no constraint on the choices of grid points and gave a recurrence relation for computing weighting coefficients of higher-order approximations. Bellman *et al.* [34] (in 1972) gave the basic idea of DQM, which was obtained by the concept of integral quadrature. After that, the above-mentioned method of finding weighting coefficients was developed by Quan and Chang in 1989 [35, 36]. A significant headway to find the weighing coefficients was made by Shu and Richards in 1990 [37].

The cubic trigonometric B-spline basis function $TB_k(x)$ is given as follows for $k=-1,2,3,\ldots\ldots,n+1$. The main idea of the present paper is to develop a new regime, a modified cubic trigonometric B-spline differential quadrature method, to acquire the numerical results of coupled the $2D$ non-linear Burgers' equation

$$TB_k(x)=\frac{1}{W}\begin{cases} r^3(x_k), & x\in[x_k,x_{k+1}],\\ r(x_k)\left[r(x_k)s(x_{k+2})+s(x_{k+3})r(x_{k+1})\right]+s(x_{k+4})r^2(x_{k+1}), & [x_{k+1},x_{k+2}],\\ s(x_{k+4})[r(x_{k+1}\,s(x_{k+3})+s(x_{k+4})r(x_{k+2})\right]+r(x_k)s^2(x_{k+3}), & [x_{k+2},x_{k+3}],\\ s^3(x_{k+4}), & [x_{k+3},x_{k+4}],\\ 0, & \text{else}\end{cases}$$

where, $r(x_k)=\sin\left(\frac{x-x_k}{2}\right)$, $s(x_k)=\sin\left(\frac{x_k-x}{2}\right)$ and $W=\sin\left(\frac{h}{2}\right)\sin(h)\sin\left(\frac{3h}{2}\right)$, and $h=\frac{b-a}{n}$.

In the proposed method, a modified cubic trigonometric B-spline has been implemented as the test function in DQM to calculate the values of weighting coefficients afterwards. The above-mentioned equation got transformed into the system of the first-order ordinary differential equations. The obtained system of equations will get solved by employing SSP-RK43 method. The accuracy and effectiveness of the proposed method will be confirmed by some test examples. Error analysis is given with the aid of L_2 and L_∞ error norms. With the aid of test examples, the exactness of the method will be checked. In Section 2, the modified cubic trigonometric B-spline DQM is implemented. In Section 2.1, a detailed discussion is given to determine the weighting coefficients at the different grid points. In Section 3, a detailed discussion is presented with the help of numerical examples. With the help of these examples, a comparison is made of the numerical and exact solutions, shown by different tables and figures. In Section 4, a brief description of the effectiveness of the proposed scheme is given.

2. Numerical scheme: modified cubic trigonometric B-spline differential quadrature method

Let us consider a one-dimension computational domain $[a,b]$ which is partitioned into N grid points s.t. $a=x_1< x_2< x_3<\ldots\ldots< x_N = b$ with a uniform step size, where, $\Delta x = x_{i+1}-x_1$. The r^{th} order derivative can be discretized with the concept of DQM such as

$$\frac{\partial^r u(x_i,\ t)}{\partial x^r}=\sum_{j=1}^{N}a_{ij}^{(r)}\,u(x_j,t)\,,\qquad i=1,2,\ .....,N\,. \tag{2.1}$$

$a_{ij}^{(r)}$ is the weighting coefficient of the r^{th} order derivative of u with respect to x. Similarly, we can continue the $1D$ discretization into $2D$ discretization. Let us consider a computational domain which is $[a,b]\times[c,d]$, where, $[a,b]$ is given with N grid points s.t. $a=x_1<x_2<x_3<\ \ldots\ldots<x_N=b$ and $c=y_1<y_2<y_3<\ \ldots\ldots<y_M=$ with uniform step sizes, given by $\Delta x=x_{i+1}-x_1$ and $\Delta y=y_{j+1}-y_j$, respectively, in the x and y directions.

By using the notion of Eq.(2.1) r^{th} spatial partial derivatives of u with respect to x (keeping y_j fixed) and y (keeping x_i fixed), respectively, can be obtained as follows

$$\frac{\partial^r u(x_i, y_j, t)}{\partial x^r} = \sum_{k=1}^{N} a_{ik}^{(r)} u(x_k, y_j, t), \qquad i = 1, 2, \ldots\ldots, N \text{ and } j = 1, 2, \ldots\ldots, M, \tag{2.2}$$

$$\frac{\partial^r u(x_i, y_j, t)}{\partial y^r} = \sum_{k=1}^{N} b_{jk}^{(r)} u(x_i, y_k, t), \qquad i = 1, 2, \ldots\ldots, N \text{ and } j = 1, 2, \ldots\ldots, M. \tag{2.3}$$

Where, $a_{ij}^{(r)}$ and $b_{ij}^{(r)}$ are known as the weighting coefficients of r^{th} order spatial partial derivatives with respect to x and y, respectively.

Similarly, r^{th} order spatial partial derivatives of $v(x, y, t)$ can be obtained with respect to x (keeping y_j fixed) and y (keeping x_i fixed), respectively, as follows:

$$\frac{\partial^r v(x_i, y_j, t)}{\partial x^r} = \sum_{k=1}^{N} a_{ik}^{(r)} v(x_k, y_j, t), \qquad i = 1, 2, \ldots\ldots, N \text{ and } j = 1, 2, \ldots\ldots, M, \tag{2.4}$$

$$\frac{\partial^r v(x_i, y_j, t)}{\partial y^r} = \sum_{k=1}^{N} b_{jk}^{(r)} v(x_i, y_k, t), \qquad i = 1, 2, \ldots\ldots, N \text{ and } j = 1, 2, \ldots\ldots, M. \tag{2.5}$$

In a similar approach, first and second order spatial partial derivatives of u with respect to x can be obtained as follows:

$$\frac{\partial u(x_i, y_j, t)}{\partial x} = \sum_{k=1}^{N} a_{ik}^{(1)} u(x_k, y_j, t), \qquad i = 1, 2, \ldots\ldots, N, \tag{2.6}$$

$$\frac{\partial^2 u(x_i, y_j, t)}{\partial x^2} = \sum_{k=1}^{N} a_{ik}^{(2)} u(x_k, y_j, t), \qquad i = 1, 2, \ldots\ldots, N. \tag{2.7}$$

In a similar approach, first and second-order spatial partial derivatives of u with respect to y respectively can be obtained as follows:

$$\frac{\partial u(x_i, y_j, t)}{\partial y} = \sum_{k=1}^{N} b_{jk}^{(1)} u(x_i, y_k, t), \qquad j = 1, 2, \ldots\ldots, M, \tag{2.8}$$

$$\frac{\partial^2 v(x_i, y_j, t)}{\partial y^2} = \sum_{k=1}^{N} b_{jk}^{(2)} v(x_i, y_k, t), \qquad j = 1, 2, \ldots\ldots, M. \tag{2.9}$$

Similarly, first and second-order spatial partial derivatives of v with respect to x are given as follows:

$$\frac{\partial v(x_i, y_j, t)}{\partial x} = \sum_{k=1}^{N} a_{ik}^{(1)} v(x_k, y_j, t), \qquad i = 1, 2, \ldots\ldots, N, \tag{2.10}$$

$$\frac{\partial^2 v\left(x_i, y_j, t\right)}{\partial x^2} = \sum_{k=1}^{N} a_{ik}^{(2)}\, v\left(x_k, y_j, t\right), \qquad i = 1, 2, \ldots, N. \tag{2.11}$$

Similarly, first and second-order spatial partial derivatives of v with respect to y are given as follows:

$$\frac{\partial v\left(x_i, y_j, t\right)}{\partial y} = \sum_{k=1}^{N} b_{jk}^{(1)}\, v\left(x_i, y_k, t\right), \qquad j = 1, 2, \ldots, M, \tag{2.12}$$

$$\frac{\partial^2 v\left(x_i, y_j, t\right)}{\partial y^2} = \sum_{k=1}^{N} b_{jk}^{(2)}\, v\left(x_i, y_k, t\right), \qquad j = 1, 2, \ldots, M. \tag{2.13}$$

Where, the selected basis is given as follows $\{TB_0(x),\ TB_1(x),\ TB_2(x),\ \ldots\ldots,\ TB_{N+1}(x)\}$ in the computational domain $\sigma = \{(x, y):\ a \le x \le b \text{ and } c \le y \le d\}$. When the 4^{th} order cubic Hyperbolic B-spline is implemented at different node points, then following the table will be obtained.

Table 1. Values at different node points.

	x_i	x_{i+1}	x_{i+2}	x_{i+3}	x_{i+4}
$TB_i(x)$	0	m_1	m_2	m_1	0
$TB_i'(x)$	0	m_3	0	m_4	0

Where

$$m_1 = \frac{\sin^2\left(\frac{h}{2}\right)}{\sin(h)\sin\left(\frac{3h}{2}\right)}, \quad m_2 = \frac{2}{1 + 2\cos(h)}, \quad m_3 = -\frac{3}{4\sin\left(\frac{3h}{2}\right)}, \quad m_4 = \frac{3}{4\sin\left(\frac{3h}{2}\right)}. \tag{2.14}$$

In order to improve the results, the modified cubic trigonometric B-spline can be implemented, in such a way so that the obtained matrix system will become diagonally dominant [38], where by using the following set of equations improvised values can be obtained.

$$\begin{aligned} &MTB_1(x) = TB_1(x) + 2TB_0(x), && MTB_{N-1}(x) = TB_{N-1}(x) - TB_{N+1}(x), \\ &MTB_2(x) = TB_2(x) - TB_0(x), && MTB_N(x) = TB_N(x) + 2TB_{N+1}(x), \\ &MTB_j(x) = TB_j(x), && (j = 3, 4, 5, \ldots, N-2). \end{aligned} \tag{2.15}$$

Where, $\{MTB_1, MTB_2, MTB_3, \ldots, MTB_N\}$ construct a basis in the computational domain.

$$\sigma = \{(x, y) : a \le x \le b \text{ and } c \le y \le d\}.$$

2.1. Determination of weighting coefficients

On substituting the values of modified trigonometric B-splines in Eq. (2.6), we will get a system of linear equations such as

$$MTB_m'(x_i)=\sum_{k=1}^{N} a_{ik}^{(1)} MTB_m(x_k), \qquad i=1,2,3,.....,N . \tag{2.16}$$

By using the tabular values in the formulae of the modified basis, a tridiagonal system of equations will be obtained as follows

$$A \ \vec{a}^{(1)}[i]=\vec{R}[i] \qquad \text{where} \qquad i=1,2,3,.....,N . \tag{2.17}$$

Where,

$$\vec{a}^{(1)}[i]=\begin{pmatrix} a_{i,1}^{(1)} \\ a_{i,2}^{(1)} \\ a_{i,3}^{(1)} \\ : \\ : \\ a_{i,\ N-1}^{(1)} \\ a_{i,N}^{(1)} \end{pmatrix}.$$

Which is known as the vector of weighting coefficients corresponding to x_i,

$$\vec{R}[i]=\begin{pmatrix} MTB_1'(x_i) \\ MTB_2'(x_i) \\ MTB_3'(x_i) \\ : \\ : \\ MTB_{N-1}'(x_i) \\ MTB_N'(x_i) \end{pmatrix}, \qquad i=1,2,3,.....,N ,$$

$$\vec{R}[1]=\begin{pmatrix} -2m_4 \\ m_3-m_4 \\ 0 \\ : \\ : \\ 0 \\ 0 \end{pmatrix},\ \vec{R}[2]=\begin{pmatrix} m_4 \\ 0 \\ m_3 \\ : \\ : \\ 0 \\ 0 \end{pmatrix},\ldots,\vec{R}[n]=\begin{pmatrix} 0 \\ 0 \\ 0 \\ : \\ : \\ m_4-m_3 \\ 2m_3 \end{pmatrix},$$

and the corresponding coefficient matrix is given as follows

$$A=\begin{pmatrix} m_2+2m_1 & m_1 & & & & 0 \\ 0 & m_2 & m_1 & & \cdots & 0 \\ & m_1 & m_2 & m_1 & & 0 \\ & \vdots & & & \ddots & \vdots \\ & & & & & 0 \\ & & & & \cdots & 0 \\ & & & & m_1 & m_2+2m_1 \end{pmatrix}.$$

Here, it is obvious that the obtained coefficient matrix is invertible. Similarly, we can apply the same concept in the Eqs (2.8), (2.10) and (2.12) in order to discretize the remaining first-order spatial partial derivatives. Second and higher order partial derivatives can be obtained by using the recurrence relation [33] given as follows:

$$a_{ij}^{(r)}=r\left[a_{ij}^{(1)}\,a_{ii}^{(r-1)}-\frac{a_{ij}^{(r-1)}}{x_i-x_j}\right]\ \text{for}\ i\neq j \tag{2.18}$$

Where, $i=1,2,3,.....,N$ and $r=2,3,4,\ .....,N-1$,

$$a_{ii}^{(r)}=-\sum_{j=1,\ j\neq i}^{N}a_{ij}^{(r)}\ \text{for}\ i=j\,. \tag{2.19}$$

Similarly, the weighting coefficients $b_{ij}^{(r)}$ for second or higher-order derivatives can be obtained by the following formula

$$b_{ij}^{(r)}=r\left[b_{ij}^{(1)}\,b_{ii}^{(r-1)}-\frac{b_{ij}^{(r-1)}}{y_i-y_j}\right]\ \text{for}\ i\neq j \tag{2.20}$$

where, $i=1,2,3,.....,N$ and $r=2,3,4,\ .....,N-1$,

$$b_{ii}^{(r)} = -\sum_{j=1,\ j\neq i}^{N} b_{ij}^{(r)} \text{ for } i=j. \tag{2.21}$$

By using the above formulae Eq.(1.1) can be discretized as follows:

$$\begin{aligned}\frac{\partial u(x_i,y_j,t)}{\partial t} &= -u(x_i,y_j)\sum_{k=1}^{N} a_{ik}^{(1)}\, u(x_k,y_j) - v(x_i,y_j) - \sum_{k=1}^{M} b_{jk}^{(1)}\, u(x_i,y_k) + \\ &+ \frac{1}{Re}\left[\sum_{k=1}^{N} a_{ik}^{(2)}\, u(x_k,y_j) + \sum_{k=1}^{M} b_{jk}^{(2)}\, u(x_i,y_k)\right]\end{aligned} \tag{2.22}$$

where, $(x_i,y_j) \in \Omega$, $t>0, i=1,2,3, \ldots\ldots, N$ and $j=1,2,3,\ldots\ldots, M$.
Similarly, by using the above formulae equation (1.2) can be discretized as follows:

$$\begin{aligned}\frac{\partial v(x_i,y_j,t)}{\partial t} &= -u(x_i,y_j)\sum_{k=1}^{N} a_{ik}^{(1)}\, v(x_k,y_j) - v(x_i,y_j) - \sum_{k=1}^{M} b_{jk}^{(1)}\, v(x_i,y_k) + \\ &+ \frac{1}{Re}\left[\sum_{k=1}^{N} a_{ik}^{(2)}\, v(x_k,y_j) + \sum_{k=1}^{M} b_{jk}^{(2)}\, v(x_i,y_k)\right]\end{aligned} \tag{2.23}$$

where, $(x_i,y_j) \in \Omega$, $t>0, i=1,2,3, \ldots\ldots, N$ and $j=1,2,3,\ldots\ldots, M$.
Equations (2.16) and (2.17) will be written as follows:

$$\frac{du(x_i,y_j,t)}{dt} = f_1\left(u(x_i,y_j,t)\right), \quad i=1,2, \ldots\ldots, N \quad \text{and} \quad j=1,2,\ldots\ldots, M, \tag{2.24}$$

$$\frac{dv(x_i,y_j,t)}{dt} = f_2\left(v(x_i,y_j,t)\right), \quad i=1,2, \ldots\ldots, N \quad \text{and} \quad j=1,2,\ldots\ldots, M. \tag{2.25}$$

Thereafter, using the scheme of SSP-RK43 the above set of equations can be solved. Accuracy is measured in terms of norms, i.e., L_2 and L_∞ error norms defined as follows:

$$L_2 = \sqrt{\sum_{i=0}^{N}\sum_{j=0}^{N}\left|u_{i,j}^{exact} - u_{i,j}^{computed}\right|^2}, \tag{2.26}$$

$$L_\infty = \max_{i,j}\left|u_{i,j}^{exact} - u_{i,j}^{computed}\right|. \tag{2.27}$$

3. Test examples

Test example 1

Considered the Eqs (1.1) and (1.2) with analytical solutions given by Fletcher [1] in 1983 as follows:

$$u(x,y,t)=\frac{3}{4}-\frac{1}{4\left[1+\exp\left((-4x+4y-t)\mathrm{Re}/32\right]\right.}, \tag{3.1}$$

$$v(x,y,t)=\frac{3}{4}+\frac{1}{4\left[1+\exp\left((-4x+4y-t)\mathrm{Re}/32\right]\right.}. \tag{3.2}$$

For domain $0\le x\le 1$ and $0\le y\le 1$. We can easily obtain the initial and boundary conditions from the given analytical solutions. In the following table, (Tab.2) L_2 and L_∞ errors for u-component have been discussed for the values $\upsilon=10^{-2}$, $\Delta t=0.0001$ at the time level $t=1.0$ and a comparison is made with [11] and [26].

Table 2. Comparison of L_2 and L_∞ errors for u-component.

Comparison of L_2 and L_∞ errors for *u*-component where $v=10^{-2}$ and $\Delta t=0.0001$ at time level $t=1.0$						
	Srivastava *et al.* (2013) [11]		Shukla *et al.* (2014) [26]		Present	
Grid Points	L_2	L_∞	L_2	L_∞	L_2	L_∞
4×4	8.57E-02	9.70E-02	1.64E-02	2.88E-03	1.55E-02	2.89E-02
8×8	4.94E-02	4.69E-02	1.93E-03	1.96E-04	5.27E-03	8.45E-03
16×16	1.92E-02	2.05E-02	3.95E-04	2.05E-05	1.09E-03	1.66E-03
32×32	8.68E-03	9.07E-03	8.12E-05	2.22E-06	2.98E-04	3.49E-04
64×64	–	–	1.53E-05	2.18E-07	7.94E-05	6.99E-05

On making the comparison with [11], it can be easily observed that errors got reduced, and on comparison with [26], it can be said that errors obtained are acceptable. In the following table, (Tab.3) L_2 and L_∞ errors for v-component have been discussed for the values $\upsilon=10^{-2}$, $\Delta t=0.0001$ at time level $t=1.0$.

Table 3. Comparison of L_2 and L_∞ errors for *v*-component.

Comparison of L_2 and L_∞ errors for *v*-component where $\upsilon=10^{-2}$ and $\Delta t=0.0001$ at time level $t=1.0$						
	Srivastava *et al.* (2013) [11]		Shukla *et al.* (2014) [26]		Present Method	
Grid points	L_2	L_∞	L_2	L_∞	L_2	L_∞
4×4	8.57E-02	9.70E-02	1.64E-02	2.88E-03	1.55E-02	2.89E-02
8×8	4.94E-02	4.69E-02	1.93E-03	1.96E-04	5.27E-03	8.45E-03
16×16	1.92E-02	2.05E-02	3.95E-04	2.05E-05	1.09E-03	1.66E-03
32×32	8.69E-03	9.08E-03	8.12E-05	2.22E-06	2.98E-04	3.49E-04
64×64	–	–	1.53E-05	2.18E-07	7.94E-05	6.99E-05

On comparing our results with [11], it is obvious that the obtained results are better than the previous ones, and comparing the present results with [26] it has been observed that the present results are acceptable. In Figure 1 a graphical representation of u and v components is given for grid point $=10\times10$, $\mathrm{Re}=100$, Δ

Δt=*0.0001* at the time level t =*1*. Numerical and exact solutions are in good agreement for both u and v components reflected in the following graph. In Figure 2, the graphical representation of numerical and exact values for u and v components is given for *10×10* grid points, Re =*150*, Δt=*0.0001* at time level t =*1*. For the given parameters, there is good compatibility between the numerical and exact solutions of both u and v components. In Figure 3, a graphical representation of numerical and exact solutions of u and v components is given for grid points=*20×20*, Re =*200*, Δt=*0.0001* at time level t =*1*. For u and v components, compatibility for the given parameters was obtained.

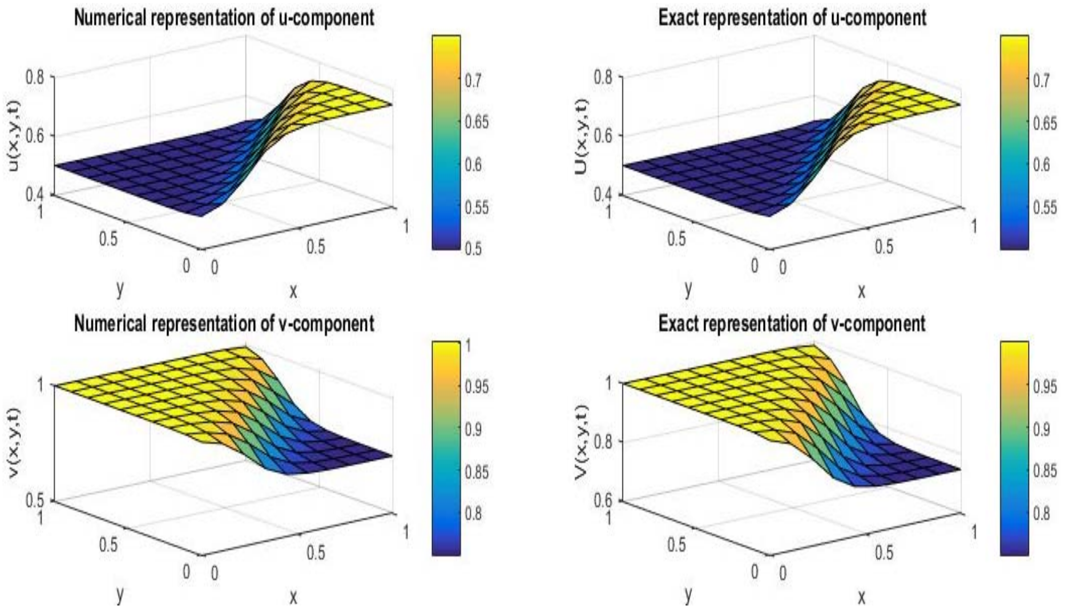

Fig.1. Graphical representation of numerical and exact values of u and v components for grid point=*10×10*, Re =*100*, Δt=*0.0001* at time level t=*1*.

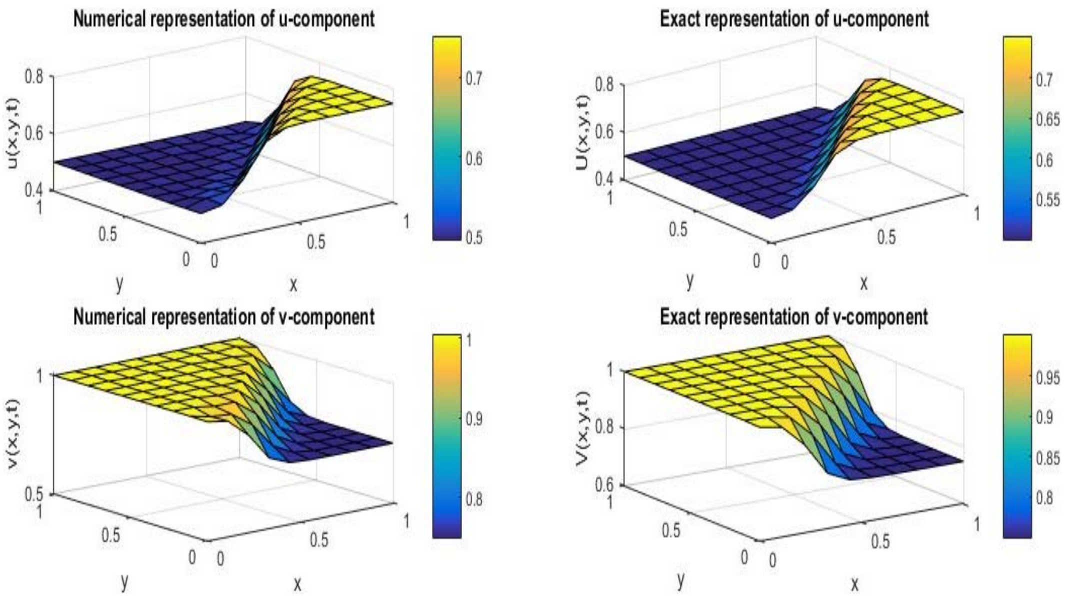

Fig.2. Graphical representation of numerical and exact values of u and v components for grid point=*10 × 10*, Re =*150*, Δt=*0.0001* at time level t=*1*.

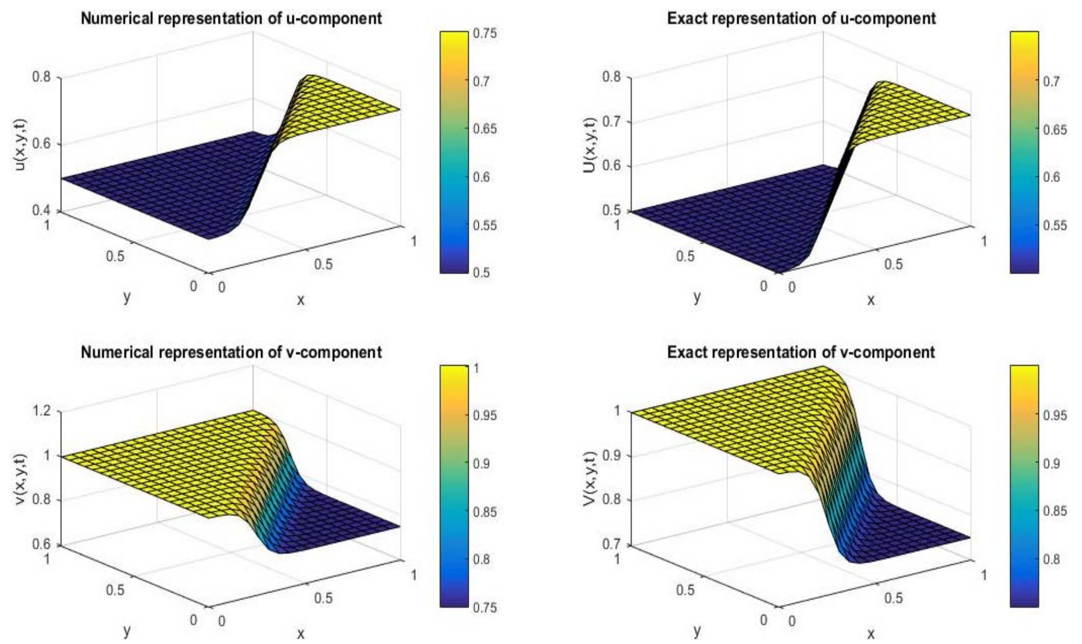

Fig.3. Graphical representation of numerical and exact values of u and v components for grid point=20×20, Re $=200$, $\Delta t=0.0001$ at time level $t=1$.

Table 4. Comparison of numerical and exact solutions of u-component.

Comparison of numerical and exact solutions of u-component at $t=0.01$ and at $t=1.0$ respectively with $\Delta t=0.0001$ and Re $=100$				
	$t=0.01$		$t=1.0$	
(x,y)	Numerical u	Exact u	Numerical u	Exact u
(0.1, 0.1)	0.623024874	0.623047034	0.510924252	0.510521932
(0.5, 0.1)	0.50124801	0.501248629	0.500059737	0.500056874
(0.9, 0.1)	0.500006486	0.500006499	0.49999987	0.500000295
(0.3, 0.3)	0.62302981	0.623047034	0.510842789	0.510521932
(0.7, 0.3)	0.501248298	0.501248629	0.500061795	0.500056874
(0.1, 0.5)	0.748671106	0.74867127	0.722944029	0.723639245
(0.5, 0.5)	0.623029807	0.623047034	0.510769687	0.510521932
(0.9, 0.5)	0.501246472	0.501248629	0.500046121	0.500056874
(0.3, 0.7)	0.748670869	0.74867127	0.723163725	0.723639245
(0.7, 0.7)	0.623029971	0.623047034	0.510759389	0.510521932
(0.1, 0.9)	0.749993096	0.749993082	0.74984181	0.749847481
(0.5, 0.9)	0.748673686	0.74867127	0.72353009	0.723639245
(0.9, 0.9)	0.623053747	0.623047034	0.510131039	0.510521932

Table 5. Comparison of numerical and exact solutions of *v*-component.

Comparison of numerical and exact solutions of *v*-component at *t=0.01* and at *t=1.0* respectively with Δ *t=0.0001* and Re =*100*				
	t=0.01		*t=1.0*	
(x, y)	numerical *v*	exact *v*	numerical *v*	exact *v*
(*0.1, 0.1*)	*0.876975126*	*0.876952966*	*0.989075748*	*0.989478068*
(*0.5, 0.1*)	*0.99875199*	*0.998751371*	*0.999940263*	*0.999943126*
(*0.9, 0.1*)	*0.999993514*	*0.999993501*	*1.00000013*	*0.999999705*
(*0.3, 0.3*)	*0.87697019*	*0.876952966*	*0.989157211*	*0.989478068*
(*0.7, 0.3*)	*0.998751702*	*0.998751371*	*0.999938205*	*0.999943126*
(*0.1, 0.5*)	*0.751328894*	*0.75132873*	*0.777055971*	*0.776360755*
(*0.5, 0.5*)	*0.876970193*	*0.876952966*	*0.989230313*	*0.989478068*
(*0.9, 0.5*)	*0.998753528*	*0.998751371*	*0.999953879*	*0.999943126*
(*0.3, 0.7*)	*0.751329131*	*0.75132873*	*0.776836275*	*0.776360755*
(*0.7, 0.7*)	*0.876970029*	*0.876952966*	*0.989240611*	*0.989478068*
(*0.1, 0.9*)	*0.750006904*	*0.750006918*	*0.75015819*	*0.750152519*
(*0.5, 0.9*)	*0.751326314*	*0.75132873*	*0.77646991*	*0.776360755*
(*0.9, 0.9*)	*0.876946253*	*0.876952966*	*0.989868961*	*0.989478068*

Table 6. Comparison of numerical and exact solutions of *u*-component.

Comparison of numerical and exact solutions of *u*-component at *t=0.01* and at *t=0.01* respectively with Δ *t=0.0001* and Re =*500*				
	t=0.01		*t=1.0*	
(x, y)	numerical *u*	exact *u*	numerical *u*	exact *u*
(*0.1, 0.1*)	*0.617976774*	*0.615254195*	*0.51889539*	*0.500000041*
(*0.5, 0.1*)	*0.500000632*	*0.5*	*0.49969085*	*0.5*
(*0.9, 0.1*)	*0.5*	*0.5*	*0.499257829*	*0.5*
(*0.3, 0.3*)	*0.617936669*	*0.615254195*	*0.506086835*	*0.500000041*
(*0.7, 0.3*)	*0.500000627*	*0.5*	*0.500039093*	*0.5*
(*0.1, 0.5*)	*0.750000011*	*0.75*	*0.756036423*	*0.749994312*
(*0.5, 0.5*)	*0.617936668*	*0.615254195*	*0.506821369*	*0.500000041*
(*0.9, 0.5*)	*0.500000802*	*0.5*	*0.49753424*	*0.5*
(*0.3, 0.7*)	*0.750000011*	*0.75*	*0.745617144*	*0.749994312*
(*0.7, 0.7*)	*0.617936671*	*0.615254195*	*0.505376292*	*0.500000041*
(*0.1, 0.9*)	*0.75*	*0.75*	*0.750000973*	*0.75*
(*0.5, 0.9*)	*0.749999942*	*0.75*	*0.751327608*	*0.749994312*
(*0.9, 0.9*)	*0.618502774*	*0.615254195*	*0.499792683*	*0.500000041*

Table 7. Comparison of numerical and exact solutions of *v*-component.

Comparison of numerical and exact solutions of *v*-component at $t=0.01$ and at $t=0.01$ respectively with $\Delta t=0.0001$ and $\mathrm{Re}=500$				
	$t=0.01$		$t=1.0$	
(x, y)	numerical *v*	exact *v*	numerical *v*	exact *v*
(0.1, 0.1)	0.882023226	0.884745805	0.98110461	0.999999959
(0.5, 0.1)	0.999999368	1	1.00030915	1
(0.9, 0.1)	1	1	1.000742171	1
(0.3, 0.3)	0.882063331	0.884745805	0.993913165	0.999999959
(0.7, 0.3)	0.999999373	1	0.999960907	1
(0.1, 0.5)	0.749999989	0.75	0.743963577	0.750005688
(0.5, 0.5)	0.882063332	0.884745805	0.993178631	0.999999959
(0.9, 0.5)	0.999999198	1	1.00246576	1
(0.3, 0.7)	0.749999989	0.75	0.754382856	0.750005688
(0.7, 0.7)	0.882063329	0.884745805	0.994623708	0.999999959
(0.1, 0.9)	0.75	0.75	0.749999027	0.75
(0.5, 0.9)	0.750000058	0.75	0.748672392	0.750005688
(0.9, 0.9)	0.881497226	0.884745805	1.000207317	0.999999959

In Table 4 numerical and exact results are given for u-component at time level $t=0.01$ and at time level $t=1.0$ respectively with $\Delta t=0.0001$ with $\mathrm{Re}=100$. For the given mesh points, a good match between the numerical and exact solutions for u-component is obtained. In Table 5, numerical and exact results are given for v-component at time level $t=0.01$ and at time level $t=1.0$, respectively, with $\Delta t=0.0001$ with $\mathrm{Re}=100$. It is obvious that a good compatibility between numerical and exact solutions is obtained for the v-component for given parameters. In Table 6, numerical and exact results are given for u-component at time level $t=0.01$ and at time level $t=1.0$, respectively, with $\Delta t=0.0001$ with $\mathrm{Re}=500$. On making a comparison between numerical and exact solutions for u-component for given parameters, it can be said that numerical results obtained are acceptable. In Table 7, numerical and exact results are given for v-component at time level $t=0.01$ and at time level $t=1.0$, respectively, with $\Delta t=0.0001$ with $\mathrm{Re}=500$. For the parameters mentioned above, a good match between numerical and exact solutions for v-component is obtained.

Test example 2

In this example, considered coupled 2D Burgers' equations have the analytical solutions as follows [31]

$$u(x,y,t)=\frac{x+y-2xt}{1-2t^2}, \tag{3.3}$$

$$v(x,y,t)=\frac{x-y-2yt}{1-2t^2}. \tag{3.4}$$

In the computational domain $0\le x\le 0.5$ and $0\le y\le 0.5$, I.C. and B.C can be easily obtained with the help of provided exact solutions. In Table 8, L_2 and L_∞ errors have been presented at different time levels for different grid points. In Table 8, L_2 and L_∞ errors are obtained for both u and v components. In Figure 4, a graphical representation of numerical and exact solutions is given for u and v components, for 5×5 grid points with $\Delta t=10^{-4}$, $\mathrm{Re}=100$ at time level $t=0.01$. For the above-mentioned parameters, a good match between numerical and exact solutions is obtained for both u and v components reflected in the graph. In Table 9, a comparison has been made between the numerical and exact solutions of u and v components at time level $t=0.1$, $N=20$, $\mathrm{Re}=100$, $\Delta t=10^{-4}$. For the given parameters, the obtained numerical results are almost the same as the exact solutions for u and v components. In Table 10, comparisons have been made of numerical results and absolute errors of u and v components with results of [29] at time level $t=0.1$, $N=20$ with $\Delta t=10^{-4}$, $\mathrm{Re}=100$. In Table 11, a comparison has been made of numerical solutions and exact solutions of u and v components at time level $t=0.4$, $N=20$, $\mathrm{Re}=100$ and $\Delta t=10^{-4}$. For components, u and v, the obtained numerical and exact solutions are in a good match. In Table 12, comparisons have been made of numerical results and absolute errors of u and v components with [29] at time level $t=0.4$, $N=20$ with $\Delta t=10^{-4}$, $\mathrm{Re}=100$.

Table 8. L_2 and L_∞ errors for *u*-component and *v*-component.

L_2 and L_∞ errors at $\Delta t=10^{-4}$, $\mathrm{Re}=100$ at time level $t=0.01$ for different grid points for *u*-component						
	at $t=0.01$		at $t=0.03$		at $t=0.05$	
Grid points	L_2	L_∞	L_2	L_∞	L_2	L_∞
5×5	3.12E-05	7.36E-05	2.86E-05	7.08E-05	2.62E-05	6.78E-05
10×10	5.99E-05	8.77E-05	5.42E-05	8.41E-05	4.88E-05	8.02E-05
15×15	7.93E-05	9.09E-05	7.11E-05	8.48E-05	6.36E-05	7.87E-05
20×20	9.48E-05	9.08E-05	8.45E-05	8.23E-05	7.54E-05	7.73E-05
25×25	1.08E-04	8.92E-05	9.59E-05	8.31E-05	8.55E-05	7.78E-05
30×30	1.20E-04	9.00E-05	1.06E-04	8.29E-05	9.46E-05	7.82E-05
35×35	1.30E-04	9.02E-05	1.15E-04	8.36E-05	1.03E-04	7.84E-05
L_2 and L_∞ errors at $\Delta t=10^{-4}$, $\mathrm{Re}=100$ at time level $t=0.01$ for different grid points for *v*-component						
	at $t=0.01$		at $t=0.03$		at $t=0.05$	
Grid points	L_2	L_∞	L_2	L_∞	L_2	L_∞
5×5	3.23E-05	7.54E-05	3.20E-05	7.64E-05	3.18E-05	7.73E-05
10×10	6.22E-05	8.85E-05	6.09E-05	8.67E-05	5.98E-05	8.42E-05
15×15	8.24E-05	8.88E-05	8.01E-05	8.79E-05	7.81E-05	8.53E-05
20×20	9.84E-05	9.10E-05	9.51E-05	8.79E-05	9.26E-05	8.72E-05
25×25	1.12E-04	9.20E-05	1.08E-04	8.90E-05	1.05E-04	8.76E-05
30×30	1.24E-04	9.14E-05	1.19E-04	8.92E-05	1.16E-04	8.76E-05
35×35	1.35E-04	9.23E-05	1.30E-04	8.92E-05	1.26E-04	8.76E-05

On making a comparison between the absolute errors for u and v components with [29], it is clear that the proposed scheme is producing better numerical approximations. In Figure 5, a graphical representation of numerical and exact solutions of u and v components is given at time level $t=0.4$, $\mathrm{Re}=200$, $N=20$ and $\Delta t=10^{-4}$. With the help of the following graph, it is obvious that the proposed scheme is giving almost the same numerical approximations as the exact solution for u and v components

Table 9. Comparison of the numerical values of *u* and *v* components.

Comparison of the numerical values of *u* and *v* components with exact values at time level *t=0.1*, *N=20* with $\Delta t = 10^{-4}$, $\mathrm{Re} = 100$				
(x, y)	numerical *u*	exact *u*	numerical *v*	exact *v*
(0.1, 0.1)	0.193327107	0.193340494	-0.021503856	-0.021482277
(0.3, 0.1)	0.365151389	0.365198711	0.193326597	0.193340494
(0.2, 0.2)	0.386653813	0.386680988	-0.043009273	-0.042964554
(0.4, 0.2)	0.558476222	0.558539205	0.17182231	0.171858217
(0.1, 0.3)	0.408157791	0.408163265	-0.279339334	-0.279269603
(0.3, 0.3)	0.579980719	0.580021482	-0.064513904	-0.064446831
(0.2, 0.4)	0.601487894	0.601503759	-0.300832497	-0.30075188
(0.3, 0.4)	0.687400447	0.687432868	-0.193421542	-0.193340494
(0.5, 0.5)	0.918367347	0.918367347	-0.102040816	-0.102040816

Table 10. Comparison of the numerical values and absolute errors.

Comparison of the numerical values and absolute errors of *u* and *v* components at time level *t=0.1*, *N=20* with $\Delta t = 10^{-4}$, $\mathrm{Re} = 100$								
Mesh point	Num. *u* [29]	Err. 1 of *u* [29]	Num. *v* [29]	Err. 2 of *v* [29]	Num. *u* [Present]	Err. 1 of *u* [Present]	Num. *v* [Present]	Err. 2 of *v* [Present]
(0.1, 0.1)	0.18368	3.31E-06	-0.02041	1.05E-06	0.193327107	1.33869E-05	-0.021503856	2.15793E-05
(0.3, 0.1)	0.34694	5.56E-06	0.18368	3.31E-06	0.365151389	4.73221E-05	0.193326597	1.38971E-05
(0.2, 0.2)	0.36735	6.62E-06	-0.04082	2.11E-06	0.386653813	2.71749E-05	-0.043009273	4.4719E-05
(0.4, 0.2)	0.53062	8.87E-06	0.16327	2.25E-06	0.558476222	6.29834E-05	0.17182231	3.59068E-05
(0.1, 0.3)	0.38776	7.67E-06	-0.26531	7.52E-06	0.408157791	5.47412E-06	-0.279339334	6.97309E-05
(0.3, 0.3)	0.55103	9.92E-06	-0.06123	3.16E-06	0.579980719	4.07632E-05	-0.064513904	6.70723E-05
(0.2, 0.4)	0.57144	1.10E-05	-0.28572	8.58E-06	0.601487894	1.58654E-05	-0.300832497	8.06177E-05
(0.3, 0.4)	0.65307	1.21E-05	-0.18368	6.40E-06	0.687400447	3.24207E-05	-0.193421542	8.10483E-05
(0.5, 0.5)	0.91838	1.65E-05	-0.10205	5.27E-06	0.918367347	0	-0.102040816	0
Where Error1=abs(Numerical u – Exact u) and Errror2=abs(Numerical v – Exact v)								

Table 11. Comparison of the numerical values of *u* and *v* components.

Comparison of the numerical values of *u* and $\boldsymbol{v}$ components and exact values at time level *t=0.4*, *N=20* with $\Delta t = 10^{-4}$, $\mathrm{Re} = 100$				
Grid points	numerical *u*	exact *u*	numerical *v*	exact *v*
(0.1, 0.1)	0.185763613	0.185758514	-0.123872186	-0.123839009
(0.3, 0.1)	0.247660739	0.247678019	0.185752661	0.185758514
(0.2, 0.2)	0.371531476	0.371517028	-0.247761363	-0.247678019
(0.4, 0.2)	0.433418941	0.433436533	0.061870515	0.061919505
(0.1, 0.3)	0.49538537	0.495356037	-0.681202511	-0.681114551
(0.3, 0.3)	0.557294786	0.557275542	-0.37162178	-0.371517028
(0.2, 0.4)	0.681147527	0.681114551	-0.805058882	-0.80495356
(0.3, 0.4)	0.712099911	0.712074303	-0.650251194	-0.650154799
(0.5, 0.5)	0.882352941	0.882352941	-0.588235294	-0.588235294

Table 12. Comparison of the numerical values and absolute errors.

Comparison of the numerical values and absolute errors with results of [29] at time level $t=0.4$, $N=20$, $\text{Re}=100$ and $\Delta t=10^{-4}$								
(x,y)	Num. u [29]	Err. 1 u [29]	Num. v [29]	Err. 2 v [29]	Num. u [Present]	Err. u [Present]	Num. v [Present]	Err. v [Present]
(0.1, 0.1)	0.17657	1.02E-04	-0.11729	3.55E-04	0.185763613	5.09935E-06	-0.123872186	3.31771E-05
(0.3, 0.1)	0.23585	5.59E-04	0.17657	1.02E-04	0.247660739	1.72795E-05	0.185752661	5.85267E-06
(0.2, 0.2)	0.35314	2.04E-04	-0.23458	7.10E-04	0.371531476	1.44477E-05	-0.247761363	8.33449E-05
(0.4, 0.2)	0.41242	6.61E-04	0.05928	4.57E-04	0.433418941	1.75917E-05	0.061870515	4.899E-05
(0.1, 0.3)	0.47044	1.51E-04	-0.64574	1.32E-03	0.49538537	2.93325E-05	-0.681202511	8.79602E-05
(0.3, 0.3)	0.52972	3.06E-04	-0.35188	1.06E-03	0.557294786	1.92445E-05	-0.37162178	0.000104752
(0.2, 0.4)	0.64701	4.90E-05	-0.76303	1.67E-03	0.681147527	3.29757E-05	-0.805058882	0.000105321
(0.3, 0.4)	0.67665	1.79E-04	-0.61611	1.55E-03	0.712099911	2.56073E-05	-0.650251194	9.63953E-05
(0.5,0.5)	0.88286	5.10E-04	-0.58646	1.77E-03	0.882352941	0	-0.588235294	0
Where Error-1=abs (Numerical u – Exact u) and Error-2=abs (Numerical v – Exact v)								

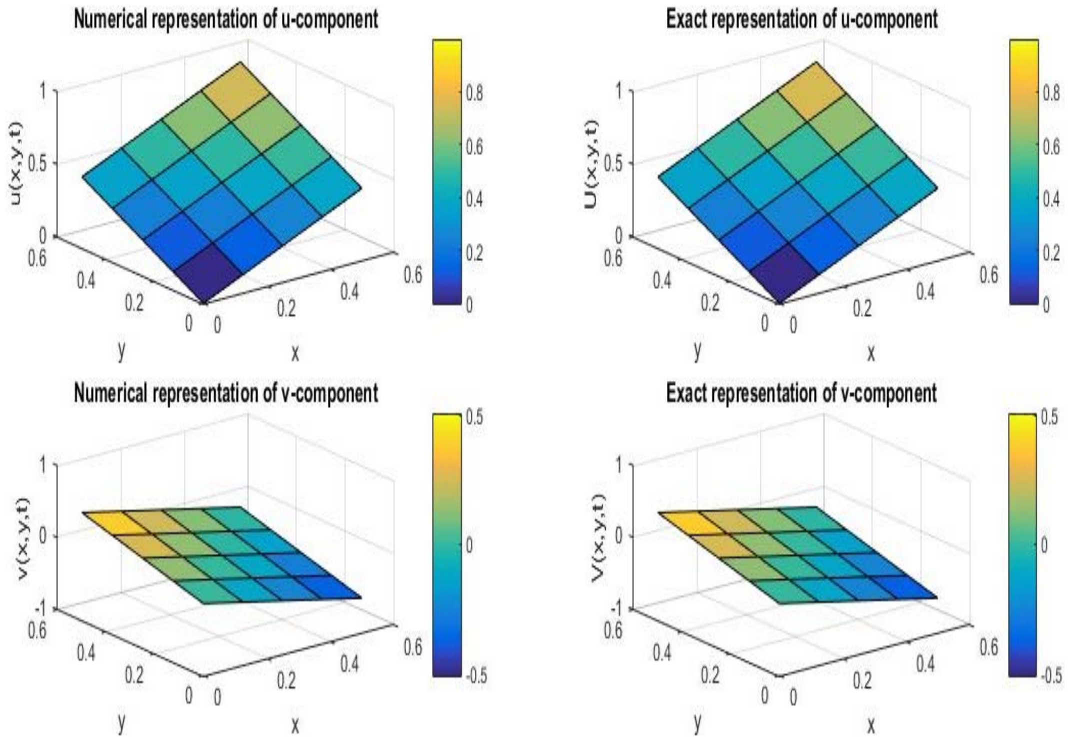

Fig.4. Comparison of numerical and exact u and v.

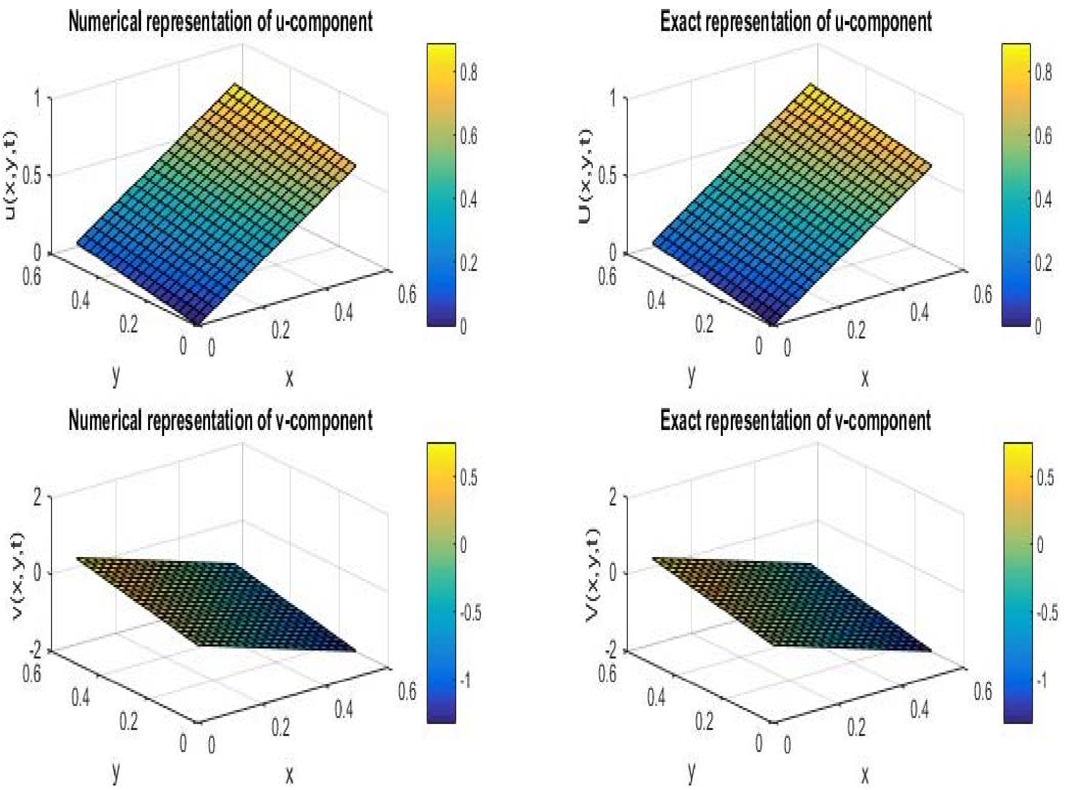

Fig.5. Comparison of numerical and exact results.

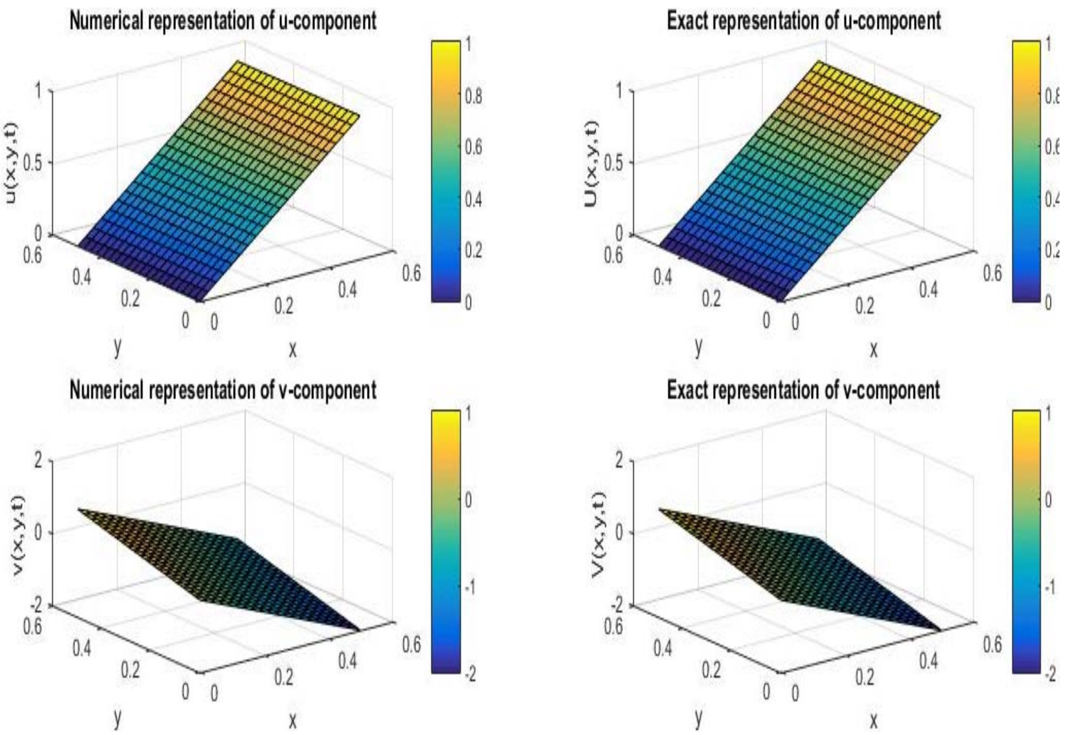

Fig.6. Comparison of numerical and exact results.

Test example 3.

In the following example [27] the Burgers' equations are given with following I.C.s and B.C.s

I.C.

$$u(x,y,0)=sin(\pi x)+cos(\pi y), \tag{3.5}$$

$$v(x,y,0)=x+y\ . \tag{3.6}$$

B.C.

$$\begin{aligned} &u(0,y,t)=cos(\pi y), && v(0,y,t)=y,\\ &u(0.5,y,t)=1+cos(\pi y), && v(0.5,y,t)=0.5+y,\\ &u(x,0,t)=1+sin(\pi x), && u(x,0,t)=x,\\ &u(x,0.5,t)=sin(\pi x), && u(x,0.5,t)=x+0.5. \end{aligned} \tag{3.7}$$

In Table 13, a comparison has been made of numerical approximations for $\mathrm{Re}=50$, $N=20$, $\Delta t=0.0001$ at the time level $t=0.625$. A comparison of numerical results for u and v components is presented and compared with, results of [11, 26, 27]. In Figure 7, a graphical representation of numerical solutions of u and v components is given for $\mathrm{Re}=50$, $\Delta t=0.0001$, $N=20$ at time level $t=0.625$. In Figure 8, a graphical representation of numerical solutions of u and v components is given for $\mathrm{Re}=100$, $\Delta t=0.0001$, $N=20$ at time level $t=0.1$. In Figure 9, a graphical representation is given for u and v components at time level $t=1$ with $\Delta t=0.0001$ and $N=20$ for $\mathrm{Re}=100$ and 200, respectively. In Figure 10, a graphical representation is given for the computed values of u and v components at time level $t=1$ with $\Delta t=0.0001$, $N=30$ for $\mathrm{Re}=150$ and 500, respectively. In Table 14, a comparison has been made of numerical values of u-component for $\mathrm{Re}=50$ at time level $t=0.625$. On making this comparison, it is quite obvious that the present scheme is producing acceptable numerical results for u-component. In Table 15, a comparison has been made of numerical values of v- component for $\mathrm{Re}=50$ at time level $t=0.625$. From the comparison of numerical solutions, it is clear that the results obtained from the presented scheme are acceptable.

Table 13. Comparison of numerical results.

Comparison of numerical results for $\mathrm{Re}=50$, grid points $N=20$, $\Delta t=0.0001$ at time level $t=0.625$								
	u-component				*v*-component			
(x,y)	[11]	[26]	[27]	Present	[11]	[26]	[27]	Present
(0.1, 0.1)	0.97146	0.97056	0.970558	0.965404126	0.09869	0.09842	0.098419	0.102491037
(0.3, 0.1)	1.15280	1.15152	1.15152	1.151264862	0.14158	0.14107	0.141070	0.14500073
(0.2, 0.2)	0.86308	0.86244	0.862434	0.851284122	0.16754	0.16732	0.167317	0.173745602
(0.4, 0.2)	0.97985	0.98078	0.980779	0.983404506	0.17111	0.17223	0.172228	0.182014656
(0.1, 0.3)	0.66316	0.66336	0.663354	0.637726918	0.26378	0.26380	0.263801	0.275178657
(0.3, 0.3)	0.77233	0.77226	0.772256	0.763142899	0.22655	0.22653	0.226526	0.236907401
(0.2, 0.4)	0.58181	0.58273	0.582728	0.571512701	0.32851	0.32935	0.329347	0.358814238
(0.4, 0.4)	0.75862	0.76179	0.761787	0.783880976	0.32502	0.32884	0.328842	0.378948318

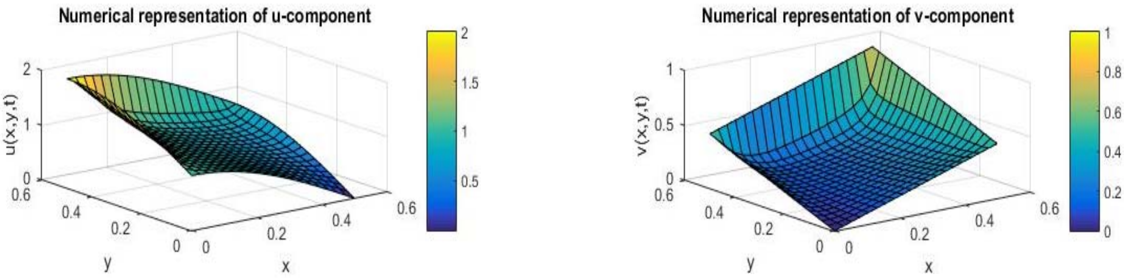

Fig.7. Numerical results of *u*-component and *v*-component.

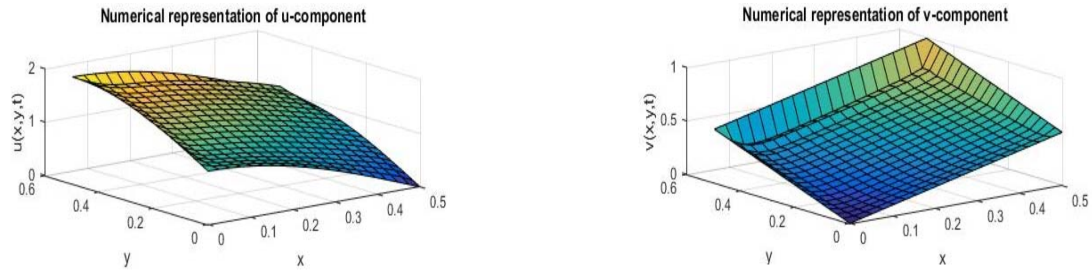

Fig.8. Numerical results of *u*-component and *v*-component.

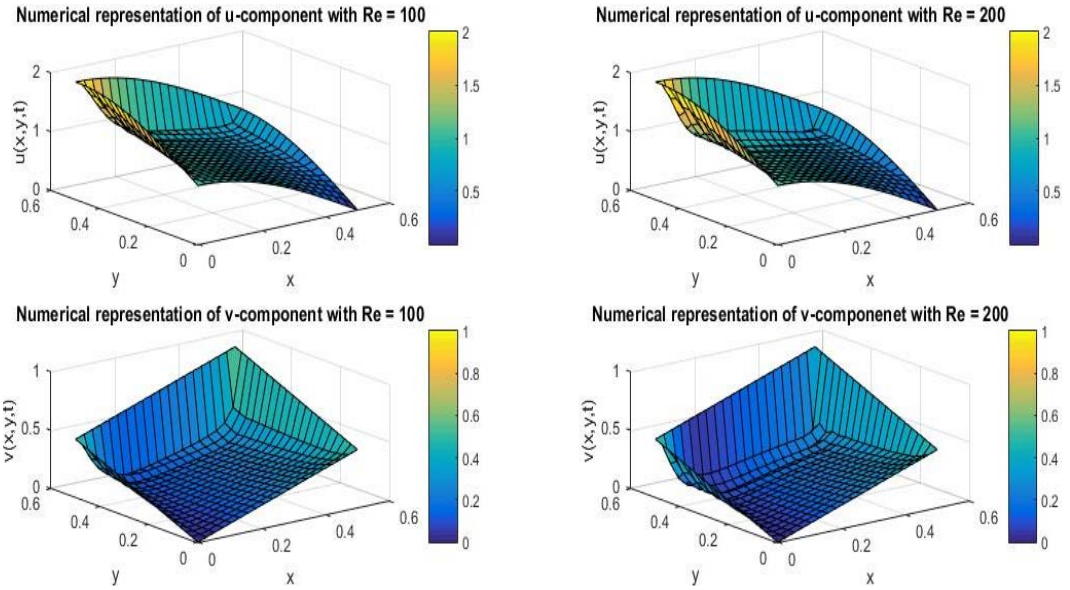

Fig.9. Comparison of numerical results.

In Table 16, a comparison has been made of numerical values of u-component for $\mathrm{Re} = 500$ at time level $t = 0.625$. A good compatibility of numerical results has been obtained for u-component. In Table 17, a comparison has been made of numerical values of v-component for $\mathrm{Re} = 500$ at time level $t = 0.625$. The obtained numerical results are acceptable for v-component.

Table 14. Comparison of numerical approximation of *u*-component.

Comparison of numerical approximation of *u*-component at time level *t=0.625* with $\mathrm{Re}=50$				
(x, y)	A.R. Bahadir [6]	Jain and Holla [2]	Srivastava and Tamsir [9]	Present Scheme
(0.1, 0.1)	0.96688	0.97258	0.97146	0.97058517
(0.3, 0.1)	1.14827	1.16214	1.1528	1.151573241
(0.2, 0.2)	0.85911	0.86281	0.86307	0.862445817
(0.4, 0.2)	0.97637	0.96483	0.97981	0.980816783
(0.1, 0.3)	0.66019	0.66318	0.66316	0.663352466
(0.3, 0.3)	0.76932	0.7703	0.7723	0.772265279
(0.2, 0.4)	0.57966	0.5807	0.5818	0.58273369
(0.4, 0.4)	0.75678	0.74435	0.75855	0.761820536

Table 15. Comparison of numerical approximation of *v*-component.

Comparison of numerical approximation of *v*-component at time level *t=0.625* with $\mathrm{Re}=50$				
(x, y)	A.R. Bahadir [6]	Jain and Holla [2]	Srivastava and Tamsir [9]	Present Scheme
(0.1, 0.1)	0.09824	0.09773	0.09869	0.09842751
(0.3, 0.1)	0.14112	0.14039	0.14158	0.141089617
(0.2, 0.2)	0.16681	0.1666	0.16754	0.167320701
(0.4, 0.2)	0.17065	0.17397	0.17109	0.172244552
(0.1, 0.3)	0.26261	0.26294	0.26378	0.263803251
(0.3, 0.3)	0.22576	0.22463	0.22654	0.22653246
(0.2, 0.4)	0.32745	0.32402	0.32851	0.329365302
(0.4, 0.4)	0.32441	0.31822	0.32499	0.328886418

Table 16. Comparison of numerical approximation of *u*-component.

Comparison of numerical approximation of *u*-component at time level *t=0.625* with $\mathrm{Re}=500$				
(x, y)	A.R. Bahadir [6]	Jain and Holla [2]	Srivastava and Tamsir [9]	Present Scheme
(0.15, 0.1)	0.9665	0.95691	0.96151	0.964459786
(0.3, 0.1)	1.0297	0.95616	1.032	1.029899051
(0.1, 0.2)	0.84449	0.84257	0.87814	0.840300536
(0.2, 0.2)	0.87631	0.86399	1.0637	0.880540396
(0.1, 0.3)	0.67809	0.67667	0.6737	0.675579736
(0.3, 0.3)	0.79792	0.76876	0.79947	0.810980759
(0.15, 0.4)	0.54601	0.54408	0.58959	0.547861661
(0.2, 0.4)	0.58874	0.58778	0.78233	0.594758021

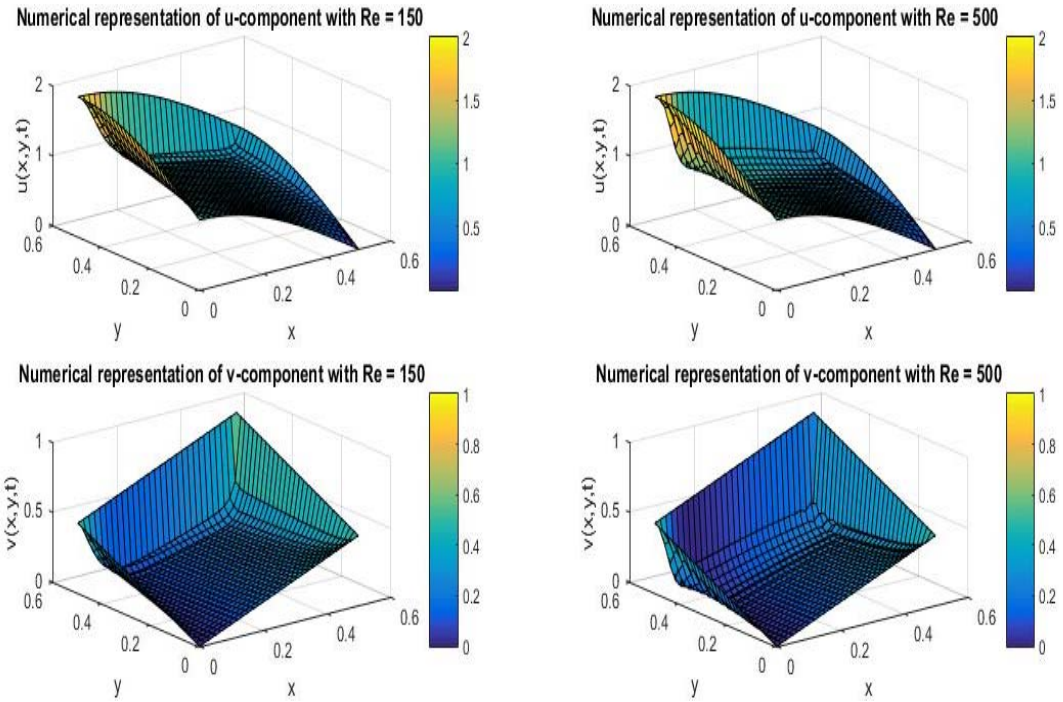

Fig.10. Comparison of numerical results.

Table 17. Comparison of numerical approximation of v-component.

Comparison of numerical approximation of v-component at time level $t=0.625$ with $\mathrm{Re}=500$				
(x, y)	A.R. Bahadir [6]	Jain and Holla [2]	Srivastava and Tamsir [9]	Present
(0.15, 0.1)	0.0902	0.10177	0.0923	0.088679672
(0.3, 0.1)	0.1069	0.13287	0.10728	0.107880524
(0.1, 0.2)	0.17972	0.18503	0.16816	0.177107115
(0.2, 0.2)	0.16777	0.18169	0.2369	0.169528037
(0.1, 0.3)	0.26222	0.2656	0.26268	0.2607306
(0.3, 0.3)	0.23497	0.25142	0.2355	0.244147695
(0.15, 0.4)	0.31753	0.32084	0.30419	0.323535367
(0.2, 0.4)	0.30371	0.30927	0.35294	0.313238238

4. Conclusion

In this paper, a modified cubic trigonometric B-spline differential quadrature method has been proposed for the numerical computation of results of the non-linear coupled 2D Burgers' equation. From the definition of DQM, the modified cubic trigonometric B-spline has been implemented as a test function in order to calculate the weighting coefficients of first-order derivative approximation. The recurrence relation given by Shu [33] is used to evaluate the weighting coefficients to approximate the second-order derivative. In Section 3, the effectiveness and compatibility of the proposed scheme are explained by introducing three test examples. The obtained results confirm that the presented scheme is producing better results. For confirming that the proposed scheme is producing better numerical approximations, the concept of L_2 and L_∞ error norms are introduced as well as comparisons are given with the aid of tables and graphs. A good compatibility between the present and existing numerical approximations and present and exact solutions is obtained. The proposed results are acceptable.

Acknowledgements

The authors thank the management of Department of Mathematics, Lovely Professional University for their continuous support to carry out this research work.

Nomenclature

Re – Reynolds number

$TB_k(x)$ – k^{th} order trigonometric B-spline

$\frac{\partial u}{\partial t}$ – first-order partial derivative of u with respect to t

$\frac{\partial u}{\partial x}$ – first-order partial derivative of u with respect to x

$\frac{\partial u}{\partial y}$ – first-order partial derivative of u with respect to y

$\frac{\partial^2 u}{\partial x^2}$ – second-order partial derivative of u with respect to x

$\frac{\partial^2 u}{\partial y^2}$ – second-order partial derivative of u with respect to y

$\frac{\partial v}{\partial t}$ – first-order partial derivative of v with respect to t

$\frac{\partial v}{\partial x}$ – first-order partial derivative of v with respect to x

$\frac{\partial v}{\partial y}$ – first-order partial derivative of v with respect to y

$\frac{\partial^2 v}{\partial x^2}$ – second-order partial derivative of v with respect to x

$\frac{\partial^2 v}{\partial y^2}$ – second-order partial derivative of v with respect to y

ν – coefficient of viscosity

Reference

[1] Fletcher C.A.J. (1983): *Generating exact solutions of the two-dimensional Burgers' equation.*– Int. J. Numer. Meth. Fluids, vol.3, pp.213-216.

[2] Jain P.C. and Holla D.N. (1978): *Numerical solution of coupled Burgers' equations.*– Int. J. of Non-Linear Mech., vol.13, pp.213-222.

[3] Fletcher A.J. (1983): *A comparison of finite element and finite difference of the one- and two dimensional Burgers' equations.*– J. Comput. Phys., vol.51, pp.159-188.

[4] Wubs F.W. and de Goede E.D. (1992): *An explicit–implicit method for a class of time-dependent partial differential equations.*– Appl. Numer. Math., vol.9, pp.157-181.

[5] Goyon O. (1996): *Multilevel schemes for solving unsteady equations*.– Int. J. Numer. Meth. Fluids, vol.22, pp.937-959.

[6] Bahadir A.R. (2003): *A fully implicit finite-difference scheme for two-dimensional Burgers' equation*.– Applied Mathematics and Computation, vol.137, pp.131-137.

[7] Srivastava V.K., Tamsir M., Bhardwaj U. and Sanyasiraju Y. (2011): *Crank-Nicolson scheme for numerical solutions of two dimensional coupled Burgers' equations*.– IJSER, vol.2, No.5, p.44.

[8] Tamsir M. and Srivastava V.K. (2011): *A semi-implicit finite-difference approach for two- dimensional coupled Burgers' equations*.– IJSER, vol.2, No.6, p.46.

[9] Srivastava V.K. and Tamsir M. (2012): *Crank-Nicolson semi-implicit approach for numerical solutions of two-dimensional coupled nonlinear Burgers' equations*.– Int. J. Appl. Mech. Eng., vol.17, No.2, pp.571-581.

[10] Srivastava V.K., Awasthi M.K. and Tamsir M. (2013): *A fully implicit Finite-difference solution to one-dimensional Coupled Nonlinear Burgers' equations*.– Int. J. Math. Sci., vol.7, No.4, p.23.

[11] Srivastava V.K., Awasthi M.K. and Singh S. (2013): *An implicit logarithmic finite difference technique for two dimensional coupled viscous Burgers' equation*.– AIP Advances, vol.3, Article ID.122105, p.9.

[12] Srivastava V.K., Singh S. and Awasthi M.K. (2013): *Numerical solutions of coupled Burgers' equations by an implicit finite-difference scheme*.– AIP Advances, vol.3, Article ID.082131, p.7.

[13] Cole J.D. (1951): *On a quasilinear parabolic equations occurring in aerodynamics*.– Q. Appl. Math., vol.9, pp.225-236.

[14] Fletcher C.A.J (1983): *A comparison of finite element and finite difference of the one and two dimensional Burgers' equations*.– J. Comput. Phys. vol.51, pp.159-188.

[15] Kutluay S., Bahadir A.R. and Ozdes A. (1999): *Numerical solution of one-dimensional Burgers' equation: explicit and exact-explicit finite difference methods*.– J. Comput. Appl. Math., vol.103, pp.251-261.

[16] Liao W (2008): *An implicit fourth-order compact finite difference scheme for one-dimensional Burgers' equation*.– Appl. Math. Comput., vol.206, pp.755-764.

[17] Ozis T., Esen A. and Kutluay S. (2005): *Numerical solution of Burgers' equation by quadratic B-spline finite elements*.– Appl. Math. Comput., vol.165, pp.237-249.

[18] Hassanien I.A., Salama A.A. , Hosham H.A. (2005): *Fourth-order finite difference method for solving Burgers' equation*.– Appl. Math. Comput., vol.170, pp.781-800.

[19] Dag I., Irk D. and Sahin A. (2005): *B-Spline collocation methods for numerical solutions of the Burgers' equation*.– Math. Probl. Eng., vol.5, pp.521-538.

[20] Korkmaz A. and Dag I. (2011): *Shock wave simulations using sinc differential quadrature method*.– Eng. Comput. Int. J. Comput. Aided Eng. Softw., vol.28, No.1, pp.654-674.

[21] Korkmaz A. and Dag I. (2011): *Polynomial based differential quadrature method for numerical solution of nonlinear Burgers' equation*.– J. Frankl. Inst., vol.348, No.10, 2863-2875.

[22] Korkmaz A., Aksoy A.M .and Dag I. (2011): *Quartic B-spline differential quadrature method*.– Int. J. Nonlinear Sci., vol.11, No.4, pp.403-411.

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

[24] Korkmaz A, and Dag I. (2012): *Cubic B-spline differential quadrature methods for the advection-diffusion equation*.– Int. J. Numer. Methods Heat Fluid Flow, vol.22, No.8, pp.1021-1036.

[25] Arora G. and Singh B.K. (2013): *Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method*.– Appl. Math. Comput., vol.224, No.1, pp.166-177.

[26] Shukla H.S., Tamsir M., Srivastava V.K .and Kumar J. (2014): *Numerical solution of two dimensional coupled viscous Burger equation using modified cubic B-spline differential quadrature method*.– AIP Adv., vol.4, Article ID.117134. p.10.

[27] Tamsir M., Srivastava V.K. and Jiwari R. (2016): *An algorithm based on exponential modified cubic B-spline differential quadrature method for nonlinear Burgers' equation*.– App. Math. and Comp., vol.290, pp.111-124.

[28] Mittal R.C. and Jiwari R. (2009): *Differential quadrature method for two-dimensional Burgers' equations*.– Int. J. for Comp. Meth. in Eng. Sci. and Mech., vol.10, No.6, pp.450-459.

[29] Zhu H., Shu H. and Ding M. (2010): *Numerical solutions of two-dimensional Burgers' equations by discrete Adomian decomposition method*.– Computers & Mathematics with Applications, vol.60, No.3, pp.840-848.

[30] Aminikhah H. (2012): A new efficient method for solving two-dimensional Burgers' equation.– ISRN Computational Mathematics, vol.2012, https://doi.org/10.5402/2012/603280

[31] Biazar J. and Aminikhah H. (2009): *Exact and numerical solutions for non-linear Burger's equation by VIM*.– Mathematical and Computer Modelling, vol.49, No.7-8, pp.1394-1400.

[32] Bert C.W. and Malik M. (1996): *Differential quadrature method in computational mechanics: a review*.– App. Mech. Rev., vol.49, No.1, p.28

[33] Shu C. (2012): *Differential Quadrature and its Application in Engineering*.– Springer Science & Business Media, p.356

[34] Bellman R., Kashef B.G. and Casti J. (1972): *Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations*.– J. Comput. Phys., vol.1, pp.40-52.

[35] Quan J.R. and Chang C.T. (1989): *New insights in solving distributed system equations by the quadrature methods-I*.– Comput. Chem. Eng., vol.13, pp.779-788.

[36] Quan J. R. and Chang C. T. (1989): *New insights in solving distributed system equations by the quadrature methods-II*.– Comput. Chem. Eng., vol.13, pp.1017–1024.

[37] Shu C. and Richards B.E. (1990): *High resolution of natural convection in a square cavity by generalized differential quadrature*.– in: Proceed. of third Conf. on Adv. Numer. Meth. Eng. Theory Appl., Swansea UK 2, pp.978-985.

[38] Arora G. and Singh B.K. (2013): *Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method*.– Applied Mathematics and Computation, vol.224, pp.166-177.

Received: February 17, 2022
Revised: May 8, 2022