Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/150215

A Novel Technique for Numerical Approximation of 2 Dimensional Non-Linear Coupled Burgers' Equations using Uniform Algebraic Hyperbolic (UAH) Tension B-Spline based Differential Quadrature Method

Varun Joshi and Mamta Kapoor*

Department of Mathematics, Lovely Professional University, Punjab, India

Received: 3 Mar. 2020, Revised: 21 Nov. 2020, Accepted: 11 Feb. 2021
Published online: 1 Mar. 2021

Abstract: A Uniform Algebraic Hyperbolic (UAH) tension B-spline of fourth order based Differential quadrature method is developed to tackle the non-linear coupled 2D Burgers' equation, to fetch the weighting coefficients, used in differential quadrature method, UAH tension B-spline is used as the basis function. UAH tension B-spline based DQM is implemented in the case of spatial discretization and in time discretization, SSP-RK 43 scheme has been implemented. By employing the developed technique upon the coupled 2D non-linear Burgers' equation, system of ODE got obtained, it is solved using SSP-RK43 scheme. In the present paper five numerical examples are tested to check the accuracy and efficiency of the developed scheme. A comparison is given in the form of L_2 and L_∞ error norms with the existing numerical schemes. Accuracy and effectiveness of developed scheme is shown in the form of tables and figures, as well. Stability of the present scheme is also tested by employing the matrix stability analysis method, which triggered the point that the developed scheme is unconditionally stable.

Keywords: Differential quadrature method, Uniform algebraic hyperbolic (UAH) tension B-spline, L_2 and L_∞ error norms, SSP-RK43 scheme, Matrix stability analysis method.

Nomenclature:
$\frac{\partial u}{\partial x}$ – *First order partial derivative of u w.r.t. x*
$\frac{\partial u}{\partial y}$ - *First order partial derivative of u w.r.t. y*
$\frac{\partial^2 u}{\partial x^2}$ – *Second order partial derivative of u w.r.t. x*
$\frac{\partial^2 u}{\partial y^2}$ – *Second order partial derivative of u w.r.t. y*
$\frac{\partial v}{\partial t}$ - *First order partial derivative of v w.r.t. t*
$\frac{\partial v}{\partial x}$ - *First order partial derivative of v w.r.t. x*
$\frac{\partial v}{\partial y}$ - *First order partial derivative of v w.r.t. y*
$\frac{\partial^2 v}{\partial x^2}$ – *Second order partial derivative of v w.r.t. x*
$\frac{\partial^2 u}{\partial y^2}$ – *Second order partial derivative of v w.r.t. y*
Re – Reynolds Number
ν = Coefficient of viscosity
R.O.C. = Rate of convergence

1 Introduction

The coupled 2D Burgers' equation is known as a simplified form of the incompressible Navier-Stokes equation. Burgers' equation is an important partial differential equation from the aspects of various physical applications, traffic flow, shock waves theory, investigating the shallow water waves, in the examination of chemical reaction-diffusion model of Brusselator and many more. Burgers' equation helps in testing a variety of Numerical Algorithms. Bateman [1] in 1915 made the first attempt to get the analytical solution of Burgers' equation and gave the derivation and steady state solution of 1D Burgers' equation. In 1950, it was used by Burgers' [2] to model the notion of turbulence. Considered 2D non-linear unsteady coupled viscous Burgers' equation is given as follows:

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=\frac{1}{Re}[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}] \quad (1)$$

* Corresponding author e-mail: varunjoshi20@yahoo.com, mamtakapoor.78@yahoo.com

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=\frac{1}{Re}[\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}] \quad (2)$$

Where initial conditions are given as follows:

$$\begin{cases} u(x,y,0)=\phi_1(x,y) \\ v(x,y,0)=\phi_2(x,y) \end{cases} \quad (3)$$

Where (x, y) $\in \Omega$
Where Ω is the computational domain as follow:

$$\Omega=(x,y)a\le x\le b, c\le y\le d \quad (4)$$

Corresponding Boundary conditions are

$$\begin{cases} u(x,y,t)=\psi_1(x,y,t) \\ v(x,y,t)=\psi_2(x,y,t) \end{cases} \quad (5)$$

Where (x, y) $\in \partial\Omega$ and t > 0, and $\partial\Omega$ is the boundary of the 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 2-D. 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 non-linear convection term,
$\nu[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}]$
is known as diffusion term, ν is viscosity coefficient greater than 0, Re is known as Reynolds number. First Fletcher [3] used the Hopf-Cole transformation to find the analytical solution of coupled 2D Burgers' equation. The numerical solution of coupled Burgers' equation is obtained by many researcher due to its importance in various fields of science and engineering. Tamsir et al. [4] used the concept of extended modified cubic B-spline DQM to approximate the solution of coupled 2D Burgers' equation. Moreover, extended modified cubic B-spline DQM was used in space, strong stability preserving Runge-Kutta stages 5 and order 4 (SSP-RK 54) was implemented in time, stability analysis of the method was also provided. Tamsir et al. [5] employed the technique of DQM built by exponential modified cubic B-spline for the solution of coupled 2D non-linear Burgers' equation and also provided the stability analysis of the matrix stability analysis method. Shukla et al. [6] solved the coupled 2D viscous non-linear Burgers' equation using the technique of MCB-DQM and the reduced set of ODE was solved by SSP-RK54 scheme. Abazari and Borhanifar [7] employed the notion of Differential Transform Method to obtain the numerical solution of coupled 2D non-linear Burgers' equation and compared with the results obtained by Homotopy Purturbation Method, Homotopy Analysis Method and the Variational Iteration Method. Mittal and Jiwari [8] implemented the concept of DQM to solve non-linear coupled 2D viscous Burgers' equation and mentioned that the results are better at Chebyshev-Gauss-Lobatto grid points compared to the uniform grid points, as well as provided the stability and convergence of the method. Bahadir [9] introduced the technique of a fully implicit finite-difference scheme to solve coupled 2D non-linear Burgers' equation. Jain and Holla [10] employed the technique of cubic spline function to solve coupled Burgers' equation. Zhu et al. [11] implemented Adomian decomposition technique to obtain the numerical solution of coupled 2D Burgers' equation. Fan et al. [12] implemented a new technique, mesh free technique, a combination of the local radial basis function (RBF) based Collocation method and the fictitious time integration method (FTIM) to solve the coupled 2D non-linear Burgers' equation. Also, the LRBFCM was adopted for the spatial discretization and the implicit Euler method was adopted for the temporal discretization. Majid Khan [13] employed the technique of the Laplace Decomposition Method (LDM) to solve the non-linear coupled 2D Burgers' equation. Hizel and Kucukarslan [14] implemented the technique of Homotopy Perturbation Method to solve $(2+1)$ D coupled Burgers' equation. Srivastava et al. [15] adopted the technique of an implicit logarithmic finite difference (I-LFDM) technique for coupled 2D viscous Burgers' equation. Srivastava and Tamsir [16] used Crank-Nicolson semi-implicit approach to obtain the numerical solution of coupled non-linear Burgers' equation. Zhanlav et al. [17] proposed a higher-order accurate explicit FD scheme to solve the 2D heat equation, which has a fourth order approximation in the space variables and a second order approximation in the time variable. They also developed a scheme to solve coupled 2D Burgers' equation.

The basic notion of DQM taken by the idea of integral quadrature and Differential quadrature method, was first introduced by Bellman et al. in 1972 [18]. After that, the above-mentioned method of finding weighting coefficients was enhanced by Quan and Chang in 1989 [19][20]. A major headway to find the weighing coefficients was given by Shu and Richards in 1990 [21]. The above-mentioned methods to determine the weighting coefficients are generalized under the exploration of higher order approximation by Shu in 1991 [22]. So far different test functions, like Lagrange polynomials, Legendre polynomial and B-spline basis functions have been used to generate different types of DQMs. Civalik [23] compared DQM and Harmonic DQM for buckling analysis of thin isotropic plates and elastic columns by . A quintic B-Spline based DQM to solve fourth order differential equation was presented by Zhong [24]. Whereas Zhong and Lan [25] proposed spline based DQM to solve the non-linear initial value problems. DQM based on sinc functions was used by Korkmaz and Dag [26]. Korkmaz and Dag [27] used Polynomial DQM to solve non-linear Burger's Equation. Quartic B-spline based DQM was introduced by Korkmaz et al. [28] to get the weighting coefficients. Arora and Singh [29] represented modified cubic B-spline based DQM (MCB-DQM) to obtain numerical solution of Burger' equation. Arora and Joshi [30] solved one and two dimensional non-linear Burger's equation using modified trigonometric cubic B-spline based DQM. Whereas Mittal and Dahiya [31] used modified cubic based B-spline as basis function in DQM to obtain the numerical solution of 3D hyperbolic equations. Mittal and Jiwari [32] used Polynomial DQM to obtain the numerical solution of non-linear Burger type equation. Jiwari et al. [33] used weighted average DQM for the solution of time dependent Burger's equation with given initial and boundary conditions. However, Shukla et al. [34] proposed exponential modified cubic B-spline based DQM (Expo-MCB-DQM) for the solution of 3D non-linear wave equation. A numerical method based upon polynomial DQM to find the numerical solution of sine-Gordon equation was presented by Jiwari et al. [35]. A numerical study using DQM of 2D reaction diffusion brusselator system was presented by Mittal and Jiwari [36]. DQM has been implemented to solve a different variety of 1D and 2D partial differential equations in the problem areas of physics, chemistry and engineering, see [37], [38], [39], [40], [32].

B-splines are vastly applied in the modeling which is free from surface and curves because of their unified mathematical properties, however, this concept involves some limitations [41], so there was a need to find some more better splines, which do not only eliminate the shortcomings of B-splines but also comprise the desired characteristics of polynomial B-splines, In recent years, a vast category of new splines have been defined in the non-polynomial space and new methods have been developed using these new kinds of B-splines. Changeable basis splines (CB splines) were introduced in [42][43]. Zhang [42] proposed the concept of C-curves in 1996, with the basis {sin(t), cos(t), t,1}. C- curves are known as the extension of cubic curves and which depend upon a parameter $\alpha > 0$. C-curves can deal with free form of curves and surfaces. It also can be used to unify the representation and processing of free and normal form curves and surfaces in engineering. Zhang [43] introduced the notion of two different forms of CB splines in 1997. He proposed a new reparametrized form of CB splines. which was defined over the domain [0, 1]. From this a third form with different parameters alpha in a curve was derived. Exponential B-splines were studied in [44]. Koch and Lyche [44] proposed the concept of exponential tension B-splines. Lu et al. [45] presented the notion of uniform hyperbolic B-splines in the space spanned by $\{sinh(t), cosh(t), t^{k-3}, \ldots\ldots, t, 1\}$. They projected a new type of uniform splines which was considered of hyperbolic polynomial B-splines, formed over the space spanned by $\{sinh(t), cosh(t), t^{k-3}, t^{k-4}, \ldots\ldots, t, 1\}$ where k is an arbitrary integer greater than or equal to three, where this point indicated that Hyperbolic B-splines have most of the properties of B-splines in the polynomial space and in their paper. A subdivision formula for this new kind of curve was also presented. It has variation diminishing properties and control polygon of the subdivision converge. Mainar and Pena [46] introduced some work upon the C-Bezier basis in the space spanned by $\{sin(t), cos(t), t^{k-3}, \ldots\ldots, t, 1\}$. In the similar approach Chen and Wang [47] did work upon the C-Bezier basis in the space spanned by $\{sin(t), cos(t), t^{k-3}, \ldots\ldots, t, 1\}$. For the same space $\{sin(t), cos(t), t^{k-3}, \ldots\ldots, t, 1\}$, Non-Uniform Algebraic Trigonometric (NUAT) spline was constructed in [48]. Wang et al. [48] proposed a new kind of splines. Non-Uniform Algebraic Trigonometric B-splines (NUAT B-splines) formed over the space spanned by $\{1, t, \ldots\ldots, t^{k-3}, cos(t), sin(t)\}$, in which k was considered as an arbitrary integer and was greater than or equal to three, where this point was that NUAT B-splines have most of the properties of usual polynomial B-splines and the subdivision formula of this kind of curves was also given. A subdivision regime was given in [49][50] based upon the concept of trigonometric splines. Jena et al. [49], presented the notion of a subdivision algorithm for the evaluation of trigonometric spline curves with uniform knot sequence in 2002. Jena et al. [50], presented a non-stationary subdivision scheme for the generalized trigonometric spline surfaces to the arbitrary mesh in 2003. Later on Hyperbolic splines were extended to the case of non-uniform knot sequence in [51], which is known as AH splines. Different types of splines over the different spaces have been proposed. Each category has its own merits. Zhang et al. [52] worked upon the uniform form of B-splines over a certain common space and then by extending the given calculation to the complex numbers, a new form FB splines with changed parameters were projected in [52]. Zhang and Krause [53] unified the concept CB splines and HB splines into FB splines. They extended the cubic uniform B-spline by unified trigonometric and hyperbolic basis, in mentioned paper, the trigonometric basis $\{sin(t), cos(t), t, 1\}$ and hyperbolic basis $\{sinh(t), cosh(t), t, 1\}$ using the shape parameter C which yields the concept of Functional B-spline (FB spline) and its corresponding subdivision B-spline. Wang and Fang [54] introduced a frequency sequence and defined unified and extended (UE) splines by an integral method over the space spanned by $\{cos(\omega t), sin(\omega t), 1, t, \ldots, t^k, \ldots\}$. UE splines unify the above-mentioned splines, which contains the desired properties of the traditional polynomial B-splines. Xu and Wang [55] proposed the notion of Algebraic Hyperbolic Trigonometric (AHT) Bezier curves and Non-uniform Algebraic Hyperbolic Trigonometric (NUAHT) B-spline curves in the space spanned by $\{sin(t), cos(t), sinh(t), cosh(t), \ldots, t^{n-5}, \ldots\ldots, t, 1\}$, $n \geq 5$. In [56], using the concept of B-splines and [56], a new tension B-spline method was developed over the space spanned by $\{sin(\tau t), cos(\tau t), sinh(\tau t), cosh(\tau t), \ldots\ldots, 1, t, \ldots\ldots, t^{n-5}\}$, where τ is known as the tension parameter and this method was based upon a non-polynomial tension B-spline function which had both trigonometric and Hyperbolic properties as well as a polynomial part.

We have considered the concepts presented in [54][56][57] to implement the notion of UAH tension B-splines with DQM. A notion of tension B-spline i.e. Uniform Algebraic hyperbolic (UAH) tension B- spline, is discussed here. Considered region here is from a to b which is divided into a mesh having uniform length $h = \frac{(b-a)}{N}$ by knots $x_i = a + ih$, i = 0, 1, 2, 3, 4, ….., N and where $a = x_0 < x_1 < x_2 < \ldots\ldots\ldots\ldots < x_N = b$.

UAH tension B-spline having order 2 is defined as follows:

$$B_{i,2}(x) = \begin{cases} \frac{sinh[\tau(x-x_{i-2})]}{sinh(\tau h)}, [x_{i-2}, x_{i-1}] \\ \frac{sinh[\tau(x_i-x)]}{sinh(\tau h)}, [x_{i-1}, x_i] \\ 0, elsewhere \end{cases} \quad (6)$$

Where $\tau = \sqrt{\eta}$ (η is a real number) For $k \geq 3$, recurrence relation of $UAHB_{i,k}$ is mentioned as follows:

$$B_{i,k}(x) = \int_{\infty}^{x} [\delta_{i,k-1} B_{i,k-1}(x) - \delta_{i+1,k-1} B_{i+1,k-1}(x)]dx \quad (7)$$

and value of $\delta_{i,j}$ is provided as follows:

$$\delta_{i,j} = (\int_{-\infty}^{\infty} B_{i,j}(x)dx)^{-1} \quad (8)$$

UAH tension B spline of third order is mentioned as follows:

$$B_{i,3}(x) = \begin{cases} [\frac{2\delta_{i,2}}{\tau sinh(\tau h)}][sinh^2\{\frac{\tau(x-x_{i-2})}{2}\}], [x_{i-2}, x_{i-1}] \\ 1 - [\frac{2\delta_{i,2}}{\tau sinh(\tau h)}][sinh^2\{\frac{\tau(x-x_i)}{2}\}] - \\ [\frac{2\delta_{i+1,2}}{\tau sinh(\tau h)}][sinh^2\{\frac{\tau(x-x_{i-1})}{2}\}], [x_{i-1}, x_i] \\ [\frac{2\delta_{i+1,2}}{\tau sinh(\tau h)}][sinh^2\{\frac{\tau(x-x_{i+1})}{2}\}], [x_i, x_{i+1}] \\ 0, otherwise \end{cases} \quad (9)$$

The present paper is organized, as follows:
In Section 2, proposed methodology is discussed. In this section, derivative approximations of the first and second order, given in partial differential equation by differential quadrature method is presented. In Section 2.1, the Uniform Algebraic Hyperbolic tension B-spline of order 4 is presented and the values of mentioned B-spline at different knot points are also given in the form of table. Modified UAH tension B-spline of order 4 is implemented for the improvised results. In Section 2.2, weighting coefficients are determined using the UAH tension B-spline of order 4 at different knot points and coefficient matrix is also fetched. In Section 3, five numerical examples are discussed to check the accuracy of proposed scheme. In Section 4, stability of the proposed method is given by matrix stability analysis method. In Section 5, brief discussion as conclusion is presented.

2 Proposed Methodology:

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

$$\frac{\partial^r u(x_i,t)}{\partial x^r} = \sum_{j=1}^{N} a_{ij}^{(r)} u(x_j,t) \quad (10)$$

Where i = 1, 2, 3, ..., N

$a_{ij}^{(r)}$ is weighting coefficient of r^{th} order derivative of u w.r.t. x. Similarly we can continue the 1D discretization into 2D discretization. Let us consider that the computational domain is [a, b] × [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 $c = y_1 < y_2 < y_3 < \ldots\ldots < y_M = d$ with uniform step sizes, given by $\Delta x = x_{i+1} - x_i$ and $\Delta y = y_{j+1} - y_j$ respectively in x and y directions. Using the notion of equation (2.1) r^{th} spatial partial derivatives of u w.r.t. x (keeping y_j fixed) and y (keeping x_i fixed), respectively, can be obtained as follows:

$$\frac{\partial^r u(x_i,y_j,t)}{\partial x^r} = \sum_{k=1}^{N} a_{ik}^{(r)} u(x_k,y_j,t) \quad (11)$$

Where i = 1, 2, 3, ..., N and j = 1, 2, 3, 4, ..., M

$$\frac{\partial^r u(x_i,y_j,t)}{\partial y^r} = \sum_{k=1}^{N} b_{jk}^{(r)} u(x_i,y_k,t) \quad (12)$$

Where i = 1, 2, 3, ..., N and j = 1, 2, 3, 4, ..., M

Where $a_{ij}^{(r)}$ and $b_{ij}^{(r)}$ are known as the weighting coefficients of r^{th} order spatial partial derivatives w.r.t. x and y, respectively. Similarly r^{th} order spatial partial derivatives of v(x,y,t) can be obtained w.r.t. x (keeping y_j fixed) and y (keeping x_i fixed) respectively as follows:

$$\frac{\partial^r v(x_i,y_j,t)}{\partial x^r} = \sum_{k=1}^{N} a_{ik}^{(r)} v(x_k,y_j,t) \quad (13)$$

Where i = 1, 2, 3, ..., N and j = 1, 2, 3, 4, ..., M

$$\frac{\partial^r v(x_i,y_j,t)}{\partial y^r} = \sum_{k=1}^{N} b_{jk}^{(r)} v(x_i,y_k,t) \quad (14)$$

Where i = 1, 2, 3, ..., N and j = 1, 2, 3, 4, ..., M

In similar approach spatial partial derivatives of u having first and second order w.r.t. x can be obtained as follows:

$$\frac{\partial u(x_i,y_j,t)}{\partial x} = \sum_{k=1}^{N} a_{ik}^{(1)} u(x_k,y_j,t) \quad (15)$$

Where i = 1, 2, 3, ..., N

$$\frac{\partial^2 u(x_i,y_j,t)}{\partial x^2} = \sum_{k=1}^{N} a_{ik}^{(2)} u(x_k,y_j,t) \tag{16}$$

Where i = 1, 2, 3, ..., N

In a similar approach, spatial partial derivatives of u having first and second order w.r.t. y, respectively, can be obtained, as follows:

$$\frac{\partial u(x_i,y_j,t)}{\partial y} = \sum_{k=1}^{N} b_{jk}^{(1)} u(x_i,y_k,t) \tag{17}$$

Where j = 1, 2, 3, ..., M

$$\frac{\partial^2 u(x_i,y_j,t)}{\partial y^2} = \sum_{k=1}^{N} b_{jk}^{(2)} u(x_i,y_k,t) \tag{18}$$

Where j = 1, 2, 3, ..., M

Similarly spatial partial derivatives of v having first and second order w.r.t. x are given as follows:

$$\frac{\partial v(x_i,y_j,t)}{\partial x} = \sum_{k=1}^{N} a_{ik}^{(1)} v(x_k,y_j,t) \tag{19}$$

Where i = 1, 2, 3, ..., N

$$\frac{\partial^2 v(x_i,y_j,t)}{\partial x^2} = \sum_{k=1}^{N} a_{ik}^{(2)} v(x_k,y_j,t) \tag{20}$$

Where i = 1, 2, 3, ..., N
Similarly spatial partial derivatives of v having first and second order w.r.t. y are given as follows,

$$\frac{\partial v(x_i,y_j,t)}{\partial y} = \sum_{k=1}^{N} b_{jk}^{(1)} v(x_i,y_k,t) \tag{21}$$

Where j = 1, 2, 3, ..., M

$$\frac{\partial^2 v(x_i,y_j,t)}{\partial y^2} = \sum_{k=1}^{N} b_{jk}^{(2)} v(x_i,y_k,t) \tag{22}$$

Where j = 1, 2, 3, ..., M

2.1 UAH tension B-spline of order 4

Uniform Algebraic Hyperbolic tension B-spline of order 4 is defined as follows:

$$B_{i,4}(x) = \begin{cases} (1)\frac{\delta_{i,3}\delta_{i,2}}{\tau sinh(\tau h)}[(x_{i-2}-x)+\frac{[sinh(\tau(x-x_{i-2}))]}{\tau}], \\ [x_{i-2},x_{i-1}] \\ (2)\delta_{i,3}[\frac{\delta i,2}{\tau sinh(\tau h)}\{(x_{i-2}-x_{i-1}) \\ +\frac{sinh[\tau(x_{i-1}-x_{i-2})]}{\tau}\}+(x-x_{i-1}) \\ -\frac{\delta_{i,2}}{\tau sinh(\tau h)}\{(x_{i-1}-x) \\ +\frac{1}{\tau}(sinh(\tau(x-x_i))+sinh(\tau(x_i-x_{i-1})))\}- \\ \frac{\delta_{i+1,2}}{\tau sinh(\tau h)}\{(x_{i-1}-x) \\ +\frac{(sinh(\tau(x-x_{i-1})))}{\tau}\}]- \\ \frac{\delta_{i+1,3}\delta_{i+1,2}}{\tau sinh(\tau h)}\{(x_{i-1}-x)+\frac{sinh(\tau(x-x_{i-1}))}{\tau}\}, \\ [x_{i-1},x_i] \\ (3)1-\frac{\delta_{i,3}\delta_{i+1,2}}{\tau sinh(\tau h)}\{(x-x_{i+1}) \\ -\frac{sinh(\tau(x-x_{i+1}))}{\tau}\} \\ -\delta_{i+1,3}[\frac{\delta_{i+1,2}}{\tau sinh(\tau h)}\{(x_{i-1}-x_i) \\ +\frac{sinh(\tau(x_i-x_{i-1}))}{\tau}\} \\ +(x-x_i)-\frac{\delta_{i+1,2}}{\tau sinh(\tau h)}\{(x_i-x) \\ +\frac{(sinh(\tau(x-x_{i+1}))-sinh(\tau(x_i-x_{i+1})))}{\tau}\} \\ -\frac{\delta_{i+2,2}}{\tau sin(\tau h)}\{(x_i-x) \\ +\frac{sinh(\tau(x-x_i))}{\tau}\}],[x_i,x_{i+1}] \\ (4)\frac{\delta_{i+1,3}\delta_{i+2,2}}{\tau sinh(\tau h)}[(x-x_{i+2})-\frac{sinh(\tau(x-x_{i+2}))}{\tau}], \\ [x_{i+1},x_{i+2}] \\ (5)0, elsewhere \end{cases} \tag{23}$$

To improve the results, modified cubic UAH tension B-spline can be implemented, so that the obtained system of matrix will become "Diagonally dominant" [29]. Using the following set of equations, improved values can be obtained.

$$\begin{cases} \phi_1(x) = B_1(x) - 2B_0(x) \\ \phi_2(x) = B_2(x) - B_0(x) \\ \phi_j(x) = B_j(x), [j = 3,4,5,......................,N-2] \\ \phi_{N-1}(x) = B_{N-1}(x) - B_{N+1}(x) \\ \phi_N(x) = B_N(x) - 2B_{N+1}(x) \end{cases} \tag{24}$$

Where $\phi_1, \phi_2, \phi_3, \ldots\ldots\ldots, \phi_N$ construct a basis in the computational Domain $\sigma = \{(x,y) : x \in [a,b] \quad and \quad y \in [c,d]\}$

2.2 Determination of weighting coefficients

Substituting the values of modified UAH tension B-splines in equation (15), we will get a system of linear equations such as,

$$\phi_m^{(1)}(x_i) = \sum_{k=1}^{N} a_{ik}^{(1)} \phi_m(x_j) \tag{25}$$

Where ***i = 1, 2, 3, ..., N***
Using the tabular values in the formulae of modified basis, “a tridiagonal system of equations” will be obtained as follows:

$$A\mathbf{a}^{(1)}[i] = \mathbf{R}[i] \tag{26}$$

Where i = 1, 2, 3, ..., N
Where $\mathbf{a}^{(1)}[i]$ =

$$\begin{bmatrix} a_{i,1}^{(1)} \\ a_{i,2}^{(1)} \\ a_{i,3}^{(1)} \\ \vdots \\ \vdots \\ \vdots \\ a_{i,N-1}^{(1)} \\ a_{i,N}^{(1)} \end{bmatrix}$$

i = 1, 2, 3, ..., N
which is known as the vector of weighting coefficients corresponding to x_i.
Where $\mathbf{R}[i]$ =

$$\begin{bmatrix} \phi_1'(x_i) \\ \phi_2'(x_i) \\ \phi_3'(x_i) \\ \vdots \\ \vdots \\ \vdots \\ \phi_{N-1}'(x_i) \\ \phi_N'(x_i) \end{bmatrix}$$

i = 1, 2, 3, ..., N and the corresponding coefficient matrix is given as follows:

$$A = \begin{bmatrix} \phi_{1,1} & \phi_{1,2} & 0 & \cdots & \cdots & \cdots & \cdots \\ \phi_{2,1} & \phi_{2,2} & 0 & \cdots & \cdots & \cdots & \cdots \\ 0 & \phi_{3,2} & \phi_{3,4} & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & \cdots & \phi_{N-2,N-2} & \phi_{N-2,N-1} & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \phi_{N-1,N-1} & \phi_{N-1,N} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \phi_{N,N-1} & \phi_{N,N} \end{bmatrix}$$

Here, it is obvious that the obtained coefficient matrix is invertible. Similarly, we can apply the same concept in the equations (17), (19) and (21) to discretize the remaining spatial partial derivatives of first-order $\frac{\partial u}{\partial y}$, $\frac{\partial v}{\partial x}$ and $\frac{\partial v}{\partial y}$. Second and higher order partial derivatives can be obtained using the recurrence relation [58], as follows:

$$a_{ij}^{(r)} = r[a_{ij}^{(1)}a_{ii}^{(r-1)} - \frac{a_{ij}^{(r-1)}}{(x_i - x_j)}], fori \neq j \tag{27}$$

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

$$a_{ii}^{(r)} = -\sum_{j=1,j\neq i}^{N} a_{ij}^{(r)}, fori = j \tag{28}$$

Similarly, the weighting coefficients $b_{ij}^{(r)}$ for second or higher order derivatives can be obtained by following formula [58],

$$b_{ij}^{(r)} = r[b_{ij}^{(1)}b_{ii}^{(r-1)} - \frac{b_{ij}^{(r-1)}}{(x_i - x_j)}], fori \neq j \tag{29}$$

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

$$b_{ii}^{(r)} = -\sum_{j=1,j\neq i}^{N} b_{ij}^{(r)}, fori = j \tag{30}$$

Using the above formulae equations (1) and (2) can be discretized as follows:

$$\begin{aligned} \frac{\partial u(x_i,y_j,t)}{\partial t} &= -u(x_i,y_j)\sum_{k=1}^{N} a_{ik}^{(1)}u(x_k,y_j) - \\ & v(x_i,y_j)\sum_{k=1}^{M} b_{jk}^{(1)}u(x_i,y_k) + \\ & \frac{1}{Re}[\sum_{k=1}^{N} a_{(ik)}^{(2)}u(x_k,y_j) + \sum_{k=1}^{M} b_{jk}^{(2)}u(x_i,y_k)] \end{aligned} \tag{31}$$

Where $(x_i,y_j) \in \sigma, t > 0$, *i = 1, 2, 3, ..., N and j = 1, 2, 3, ..., M*

$$\begin{aligned} \frac{\partial v(x_i,y_j,t)}{\partial t} &= -u(x_i,y_j)\sum_{k=1}^{N} a_{ik}^{(1)}v(x_k,y_j) - \\ & v(x_i,y_j)\sum_{k=1}^{M} b_{jk}^{(1)}v(x_i,y_k) + \\ & \frac{1}{Re}[\sum_{k=1}^{N} a_{(ik)}^{(2)}v(x_k,y_j) + \sum_{k=1}^{M} b_{jk}^{(2)}v(x_i,y_k)] \end{aligned} \tag{32}$$

Where $(x_i,y_j) \in \sigma, t > 0$, *i = 1, 2, 3, ..., N and j = 1, 2, 3, ..., M*
Equations (31) and (32) will be written as follows:

$$\frac{\partial u(x_i,y_j,t)}{dt} = f_1(u(x_i,y_j,t)),$$
i = 1, 2, 3, ..., N and j = 1, 2, 3, ..., M

and

$$\frac{\partial v(x_i,y_j,t)}{dt} = f_2(v(x_i,y_j,t)),$$
i = 1, 2, 3, ..., N and j = 1, 2, 3, ..., M

Table 1: Tabular values of UAH tension B spline of order 4 i.e. $B_{i,4}(x)$ and $B'_{i,4}(x)$ at different knot points:

	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}
$B_{i,4}(x)$	0	a_1	a_2	a_3	0
$B'_{i,4}(x)$	0	a_4	0	a_5	0

Table 2: Comparison of L_2 and L_∞ errors for u component where $\nu = 10^{-2}$ and Δt = 0.0001 at time level t = 1.0 with τ = 0.1

	Srivastava et al. (2013) [15]		Present Method	
Grid Points	L_2	L_∞	L_2	L_∞
4×4	8.57E-02	9.70E-02	1.54E-02	2.81E-02
8×8	4.94E-02	4.69E-02	5.25E-03	8.40E-03
16 ×16	1.92E-02	2.05E-02	1.08E-03	1.65E-03

Table 3: Comparison of L_2 and L_∞ errors for v component where ν = 10^{-2} and Δt = 0.0001 at time level t = 1.0 with τ = 0.1

	Srivastava et al. (2013) [15]		Present Method	
Grid points	L_2	L_∞	L_2	L_∞
4 x 4	8.57E-02	9.70E-02	1.54E-02	2.81E-02
8 x 8	4.94E-02	4.69E-02	5.25E-03	8.40E-03
16 x 16	1.92E-02	2.05E-02	1.08E-03	1.65E-03

After that, using the scheme of SSP-RK43, the aforementioned set of equations can be solved. The measurement of accuracy is measured in terms of norms i.e. L_2 and L_∞ error norms are defined, as follows:

$$L_2 = \|u_{exact} - u_{computed}\|_2 = \sqrt{\sum_{i=0}^{N}\sum_{j=0}^{N} |u_{i,j}^{exact} - u_{i,j}^{computed}|^2}$$

$$L_\infty = \|u_{exact} - u_{computed}\|_\infty = \max_{i,j} |u_{i,j}^{exact} - u_{i,j}^{computed}|$$

3 Numerical Experiments and Discussion

In this section, five numerical examples are discussed. Accuracy of the present method is presented using the concept of L_2 and L_∞ error norms. The present numerical results are compared with the existing numerical results as well as with the exact solution. Graphical representation is also presented by means of figures.

Example 1. Considered the equations (1) and (2) with analytical solutions given by Fletcher in 1983 [3] as follows,

$$u(x,y,t) = \frac{3}{4} - \frac{1}{4\{1+exp((-4x+4y-t)\frac{Re}{32})\}}$$

$$v(x,y,t) = \frac{3}{4} + \frac{1}{4\{1+exp((-4x+4y-t)\frac{Re}{32})\}}$$

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. Regarding this Example, following points are discussed as follows. In Table 2 and Table 3, comparison of L_2 and L_∞ errors for both u and v components is done with [15] for $\nu = 10^{-2}$, Δt = 0.0001 at time level t = 1 with τ = 0.1. Obtained results are better than the compared ones. In Table 4 and Table 5, comparison of numerical results for both u and v components is made with [9] and [8] as well as with the exact solution for number of grid points 21, Re = 100, Δt = 0.001, τ = 1 at t = 0.01. In Table 6 and Table 7, comparison of numerical results is made with [9] and [8] and with the exact solution for the mentioned parameters in tables. Good match of numerical results is obtained. In Table 8 and Table 9, comparison of numerical results is presented at the time level t = 2 with [9] and [8] and with the exact solution for both u and v components for number of grid points 21, Re = 100, Δt = 0.5 and τ = 1. In Table 10 and Table 11, obtained numerical results are compared with [11] as well as with the exact solution for both u and v components at the different time levels for the given parameters. In Figure 1 and Figure 2, graphical representation of numerical and exact solutions for both u and v components is given at the time levels t = 1 and t = 0.5 respectively.

Example 2. In the present example, considered coupled 2D Burgers' equations are having the analytical solutions as follows [62]:

$$u(x,y,t) = \frac{x+y-2xt}{1-2t^2} \tag{33}$$

$$v(x,y,t) = \frac{x-y-2yt}{1-2t^2} \tag{34}$$

In the computational domain $0 \le x \le 0.5$ and $0 \le y \le 0.5$, where I.C. and B.C can be easily obtained with the help of provided exact solutions. In this example, in Table 12, comparison of absolute error is made for u component with [11] at different mesh points. It is noticeable that produced results are better than the compared ones. In Table 13, comparison of the absolute error for v component is made with [11] for the mentioned parameters in table and obtained numerical results are better than [11]. In Table 14 and Table 15, a comparison between [11] and the present results is presented for the absolute error at the time level t = 0.4, for both u and v components. Obtained results are in acceptable state. In

Table 4: Comparison of Numerical and Exact solutions for 21x21, Re = 100, Δt = 0.001, τ = 1 at t = 0.01

u component				
Mesh Point	**Bahadir [9]**	**Mittal and Jiwari [8]**	**Present method**	**Exact solution**
(0.1, 0.1)	6.23E-01	6.23E-01	0.622847504	0.623047034
(0.9, 0.1)	5.00E-01	5.00E-01	0.500010967	0.500011
(0.7, 0.3)	0.50162	5.02E-01	0.501617103	0.501622067

Table 5: Comparison of Numerical and Exact solutions for 21x21, Re = 100, Δt = 0.001, τ =1 at t = 0.01

v component				
Mesh Point	**Bahadir [9]**	**Mittal and Jiwari [8]**	**Present Method**	**Exact Solution**
(0.1, 0.1)	0.87688	0.87695	0.87715	0.87695
(0.9, 0.1)	0.99998	0.99999	0.99998	0.99999
(0.7, 0.3)	0.99838	0.99838	0.99838	0.99838

Table 6: Comparison of Numerical and Exact solutions for 21x21, Re = 100, Δt = 0.001, τ = 1 at t = 0.5

u component				
Mesh Point	**Bahadir [9]**	**Mittal and Jiwari [8]**	**Present Method**	**Exact Solution**
(0.1, 0.1)	0.54235	0.54322	0.54409	0.54332
(0.9, 0.1)	0.49931	0.5	0.5	0.5
(0.7, 0.3)	0.49961	0.50035	0.50036	0.50035

Table 16 and Table 17, comparison of numerical and exact solutions for both u and v components is presented at the time level t = 0.01 and t = 0.03 and absolute error is also mentioned. In Table 18 and Table 19, numerical and exact solutions are mentioned for the both u and v components at time levels t = 0.1 and t = 0.3 and absolute error is also included. In Figures 3, 4 and 5, a comparison between numerical and exact solutions of both u and v components are presented at the mentioned parameters.

Example 3. In the following example Burgers' equations are given with

Initial conditions:

$$u(x,y,0) = sin(\pi x) + cos(\pi y) \tag{35}$$

$$v(x,y,0) = x + y \tag{36}$$

Boundary Conditions:

$$u(0,y,t) = cos(\pi y), \tag{37}$$

$$u(0.5,y,t) = 1 + cos(\pi y), \tag{38}$$

$$u(x,0,t) = 1 + sin(\pi x), \tag{39}$$

$$u(x,0.5,t) = sin(\pi x), \tag{40}$$

$$v(0,y,t) = y, \tag{41}$$

$$v(0.5,y,t) = 0.5 + y, \tag{42}$$

$$v(x,0,t) = x, \tag{43}$$

$$v(x,0.5,t) = x + 0.5 \tag{44}$$

In Table 20 of present example, fetched numerical results are compared with [15], [6] and [5] for the u component at time level t = 0.625 and in Table 21 comparison of numerical solution is given with [15], [6] and [5] for v component at the time level t = 0.625. In Figures 6, 7, 8 and 9 a graphical representation of numerical and exact solutions is given for both u and v components for the mentioned parameters in figures.

Example 4. In this example considered coupled Burgers' equations has the exact solution [61] as follows:

$$u(x,y) = sin(\pi x)sin(\pi y) \tag{45}$$

$$v(x,y) = [sin(\pi x) + sin(2\pi x)][sin(\pi y) + sin(2\pi y)] \tag{46}$$

Initial conditions can be easily fetched from the provided exact solution and all boundary conditions are zero. In

Table 7: Comparison of Numerical and Exact solutions for 21x21, Re = 100, Δt = 0.5, τ = 1 at t = 0.5
v component

Mesh Point	Bahadir [9]	Mittal and Jiwari [8]	Present Method	Exact Solution
(0.1, 0.1)	0.95577	0.95678	0.9559	0.95667
(0.9, 0.1)	0.99861	1	0.99999	0.99999
(0.7, 0.3)	0.99827	0.99964	0.99963	0.99964

Table 8: Comparison of Numerical and Exact solutions for 21x21, Re = 100, Δt = 0.5, τ = 1 at t = 2
u component

MP	Bahadir [9]	Mittal and Jiwari [8]	Present Method	Exact Solution
(0.1, 0.1)	0.49983	0.50048	0.5005	0.50048
(0.9, 0.1)	0.4993	0.5	0.49999	0.5
(0.7, 0.3)	0.4993	0.5	0.5	0.5

Table 9: Comparison of Numerical and Exact solutions for 21x21, Re = 100, Δt = 0.5, τ = 1 at t = 2
v component

Mesh Point	Bahadir [9]	Mittal and Jiwari [8]	Present Method	Exact Solution
(0.1, 0.1)	0.99826	0.99952	0.99949	0.99951
(0.9, 0.1)	0.99861	1	1	1
(0.7, 0.3)	0.9986	1	0.99999	0.99999

this example no exact solution is given. Domain = [0,1] × [0,1]. Table 22, comparison of numerical results is made with [59] and [60] for both u and v components at the time level t = 0.01. In Table 23, a comparison of numerical result is given, for both u and v components at different mesh points, with [61]. Graphical representation is given for numerical solutions of u components at time levels t = 0.1, 0.3, 0.5 and 1 respectively in Figure 10 and numerical solution's graphical representation is given for v component at time level t = 0.1, 0.3, 0.5, 1 in Figure 11.

Example 5. In this example considered coupled equations are given with following Exact solutions [60][61].

$$u(x,y,t) = -exp{-2\nu t}sin(x+y) \tag{47}$$

$$v(x,y,t) = exp{-2\nu t}sin(x+y) \tag{48}$$

Where Domain = $[-\pi,\pi]\times[-\pi,\pi]$
In this example, Table 24 includes the description of L_2 and L_∞ errors for the both u and v components at time level t = 0.01 and t = 0.05 respectively. Errors got reduced on increasing the number of grid points. In Figure 12, numerical solution's graph for both u and v components are given at the time level t = 0.05 with the given parameters.

Order of convergence:

$$ROC = \frac{ln(\frac{E(N_2)}{E(N_1)})}{ln(\frac{N_1}{N_2})}$$

Where $E(N_j)$ denotes either L_2 error norm or L_∞error norm. Order of convergence is provided for Example 1 with the aid of Tables 25, 26, 27 and 28 for both U and V components at the time levels t = 1 and t = 2 respectively.

Comparison of RMS Error norm and Relative Error along with the time of computations:

RMS Error Norm =

$$\sqrt{\frac{\sum_{j=1}^{n}(u_{ij}-U_{ij})^2}{N\times N}}$$

and u_{ij} is the corresponding exact solutions and U_{ij} is the obtained numerical solution.
Relative Error =

$$\sqrt{\frac{\sum_{j=1}^{n}(u_{ij}-U_{ij})^2}{u_{ij}^2}}$$

Where u_{ij} is the corresponding exact solution and U_{ij} is the obtained numerical solution. In Tables 29 and 30, a comparison between RMS error norm and Relative error norm is provided along with the computational time. It can be easily observed that on increasing the number of grid points, both type of errors reduced.

Error Estimate: Let $UAHB_k(x)$ {k = -1,. . . . . . , n+1} Uniform Algebraic Hyperbolic tension B-spline be the set of basis functions, which is continuously differentiable twice and linearly independent in the given universe if discourse. It means that the cubic Uniform Algebraic Hyperbolic tension B-spline is (n+3) dimensional

Table 10: Comparison of Numerical and Exact solutions for u component with Re = 80, Δt = 10^{-4}, τ= 0.1 at the given time levels

	t = 0.05			t = 0.2			t = 0.5		
Mesh Point	**Numerical [Zhu et al] [11]**	**Present**	**Exact**	**Numerical [Zhu et al] [11]**	**Present**	**Exact**	**Numerical [Zhu et al] [11]**	**Present**	**Exact**
(0.1, 0.1)	0.61733	0.61723	0.61719	0.59465	0.59474	0.59438	0.55601	0.55624	0.55567
(0.8, 0.3)	0.50147	0.50113	0.50113	0.50098	0.50078	0.50078	0.50029	0.50038	0.50037
(0.9, 0.5)	0.50395	0.50322	0.50323	0.50266	0.50222	0.50222	0.50086	0.50106	0.50105

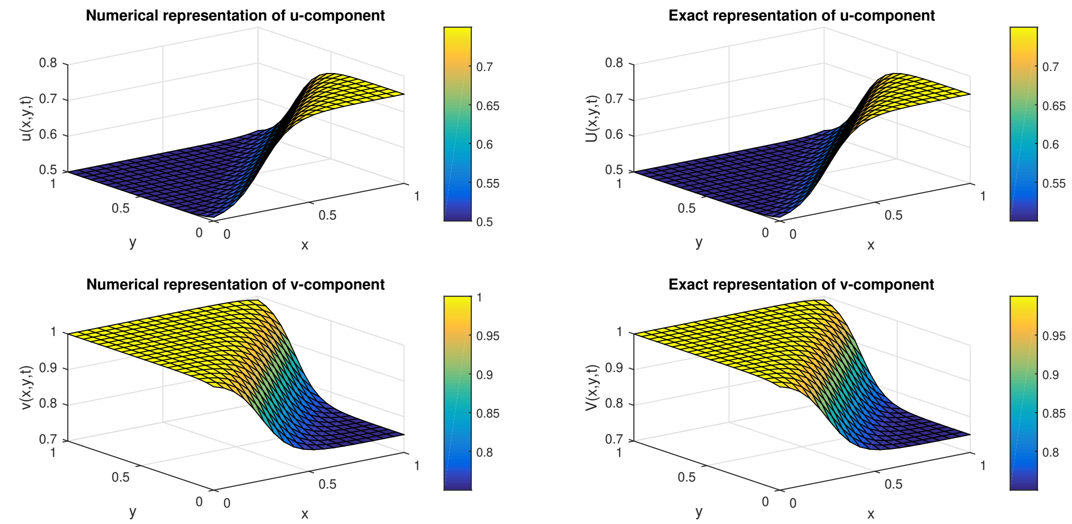

Fig. 1: Graphical representation of Numerical and Exact solutions of u and v components for Δt = 0.0001, ν = 10^{-2}, τ = 1, N = 20 at time level t = 1 for Example 1

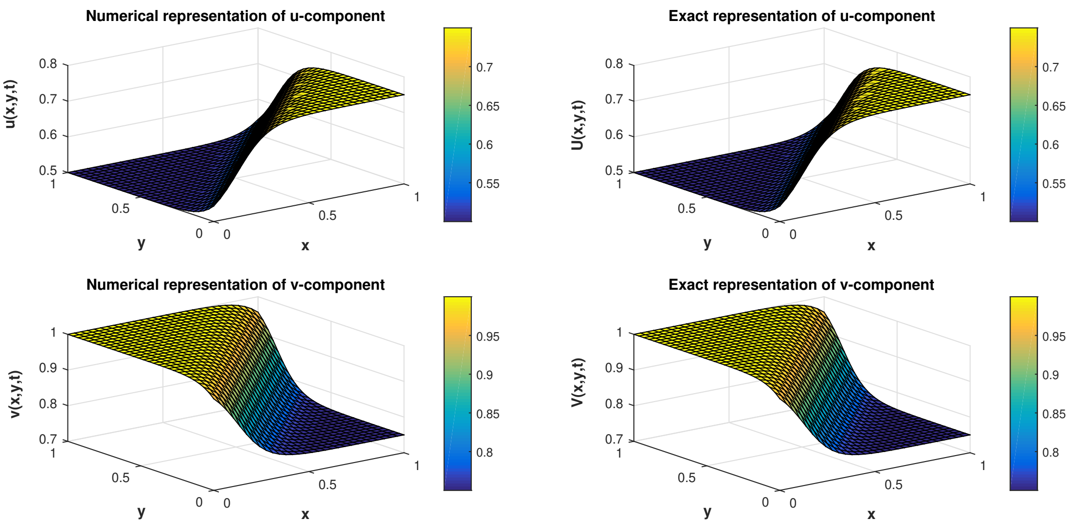

Fig. 2: Graphical representation of Numerical and Exact solutions of u and v components for Δt = 0.0001, ν = 10^{-2} , τ = 1, N = 30 at time level t = 0.5 for Example 1

Table 11: Comparison of Numerical and Exact solutions for v component with Re = 80, $\Delta t = 10^{-4}$, τ= 0.1 at the given time levels

	t = 0.05			t = 0.2			t = 0.5		
Mesh Point	**Numerical Zhu et al. [11]**	**Present**	**Exact**	**Numerical Zhu et al. [11]**	**Present**	**Exact**	**Numerical Zhu et al. [11]**	**Present**	**Exact**
(0.1, 0.1)	0.88267	0.88276	0.8828	0.90534	0.90525	0.90561	0.94399	0.94375	0.94432
(0.8, 0.3)	0.99853	0.99886	0.99886	0.99902	0.99921	0.99921	0.99971	0.99962	0.99962
(0.9, 0.5)	0.99605	0.99677	0.99676	0.99734	0.99777	0.99777	0.99913	0.99894	0.99894

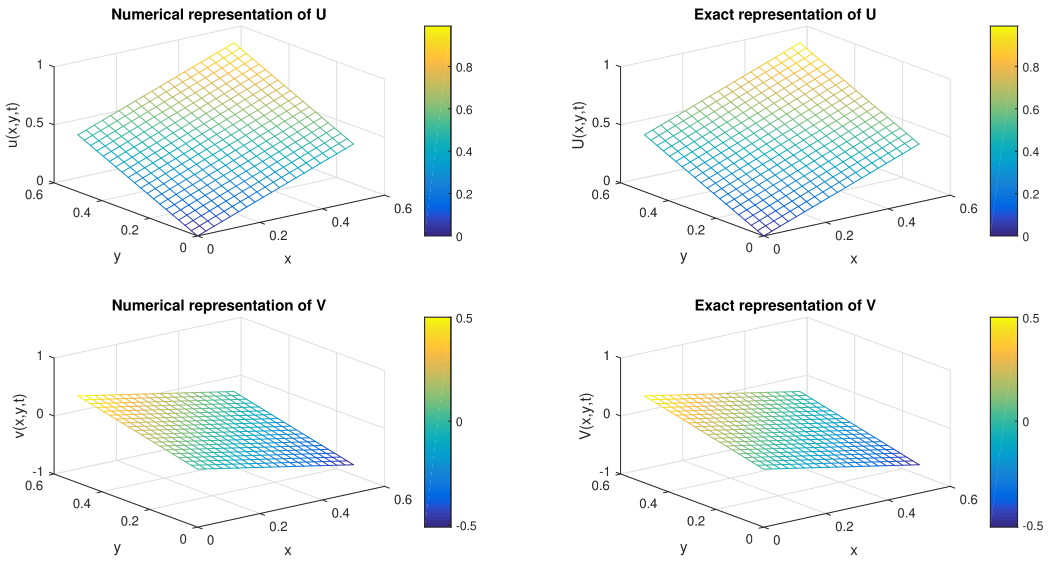

Fig. 3: Graphical representation of Numerical and Exact solutions of u and v components a time level t = 0.01, $\Delta t = 10^{-4}$, Re = 100 and τ = 1 with N = 20

Table 12: Comparison between absolute error of u component where t = 0.1, N = 20 where Reynolds number is taken arbitrarily and τ = 0.1 where Error1 = abs(Numerical u – Exact u)

Mesh	Error 1 Zhu et al. [11]	Error 1 [Present]
(0.1, 0.1)	3.31E-06	1.27E-07
(0.3, 0.1)	5.56E-06	4.35E-07
(0.2, 0.2)	6.62E-06	2.72E-07

Table 13: Comparison between absolute error of v component where t = 0.1, N = 20, with arbitrary Reynolds number and τ = 0.1 where Error 2 = abs(Numerical v – Exact v)

Mesh Point	Error 2 Zhu et al. [11]	Error 2[Present]
(0.1, 0.1)	1.05E-06	1.99E-07
(0.3, 0.1)	3.31E-06	1.45E-07
(0.2, 0.2)	2.11E-06	4.46E-07

subspace of $C^2[a,b]$. Using the notion of $UAHB_k(x)$, we will get the following formulae,

$$u^{(p)}(x_i) = \sum_{j=1}^{n} w_{ij}^{(p)} u(x_j)$$

or

$$UAHB_k'(x_i) = \sum_{j=0}^{n} w_{ij}^{(1)} UAHB_k(x_j)$$

$$UAHB_k''(x_i) = \sum_{j=0}^{n} w_{ij}^{(2)} UAHB_k(x_j)$$

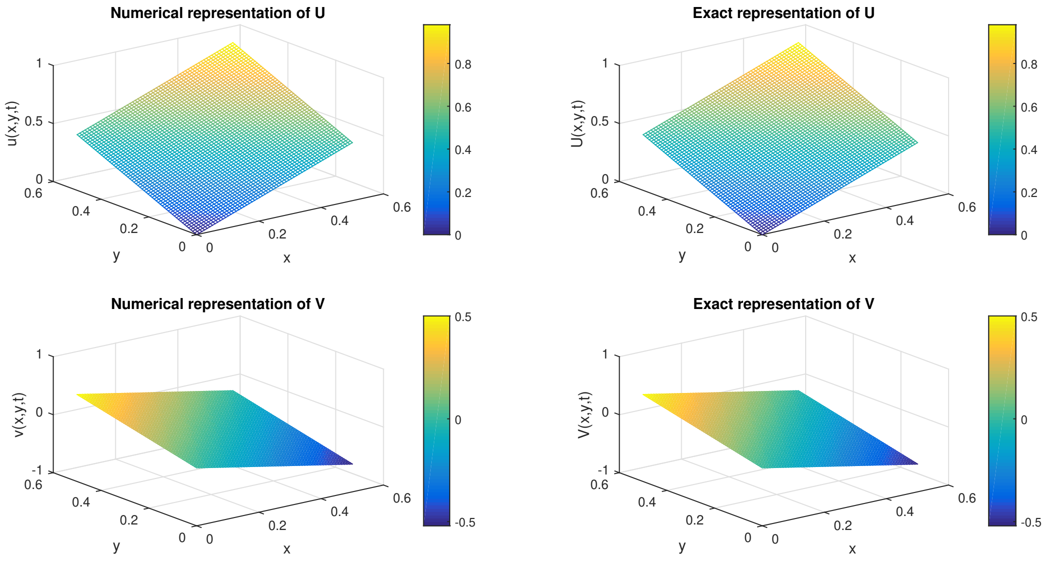

Fig. 4: Graphical presentation of Numerical and Exact u and v components at time level t = 0.02, with $\Delta t = 10^{-4}$, Re = 100, τ = 1 and N = 60

Table 14: Comparison between absolute error of u component where t = 0.4, N = 20, Reynolds number is taken arbitrarily and τ = 0.1 where Error 1 = abs(Numerical u – Exact u)

Mesh Point	Error1 Zhu et al. [11]	Error 1 [present]
(0.1, 0.1)	1.02E-04	4.49E-08
(0.3, 0.1)	5.59E-04	1.46E-07
(0.2, 0.2)	2.04E-04	1.03E-07

Table 15: Comparison between absolute error of v component where t = 0.4, N = 20, where Error 2 = abs(Numerical v – Exact v)

Mesh Point	Error 2 Zhu et al. [11]	Error 2 [present]
(0.1, 0.1)	3.55E-04	2.89E-07
(0.3, 0.1)	1.02E-04	8.28E-08
(0.2, 0.2)	7.10E-04	6.73E-07

By solving the system, $w_{ij}^{(p)}$ can be obtained and the derivative of the given function can be obtained at the node points. Cubic Uniform Algebraic Hyperbolic tension B-spline $\phi(x)$ is given as follows;

$$\phi(x) = \sum_{k=-1}^{n+1} m_k UAHB_k(x)$$

Where m_k can be determined using the interpolation conditions given as [63][64],

$$\phi(x_i) = u(x_i),\ i = 0,\ 1,\ \ldots\ldots,\ n$$

$$\phi''(a) = u''(a) - \frac{\phi^2}{12}u^{(4)}(a)$$

$$\phi''(b) = u''(b) - \frac{\phi^2}{12}u^{(4)}(b)$$

We can construct the approximation of u(x) as follows:

$$u'(x_i) \cong U'(x_i)$$

$$= \sum_{j=-1}^{n+1} w_{ij}^{(1)} U(x_j), i = -1,\ \ldots\ldots,\ n+1$$

Theorem: Let $v(x) \in C^6[a,b]$. Then the two above-mentioned approximatuons of $u'(x_i)$ and $u''(x_i)$ have the following error bounds; for i = -1,. . . . ., n+1

$$|u'(x_i) - U'(x_i)| = O(h^4)$$

$$|u''(x_i) - U''(x_i)| = O(h^4)$$

Table 16: Comparison of Numerical and Exact solutions at time levels t = 0.01 and 0.03 respectively with Δt = 0.00001, Re = 50 and τ = 0.5 for u component

	t = 0.01			t = 0.03		
Mesh Point	**Numerical u**	**Exact u**	**Abs. Error u**	**Numerical u**	**Exact u**	**Abs. Error u**
(0.1, 0.1)	2.08E-01	2.08E-01	2.02E-06	2.05E-01	2.05E-01	1.86E-06
(0.2, 0.1)	3.12E-01	3.12E-01	4.09E-06	3.04E-01	3.04E-01	3.84E-06
(0.3, 0.1)	4.15E-01	4.15E-01	6.15E-06	4.03E-01	4.03E-01	5.82E-06

Table 17: Comparison of Numerical and Exact solutions at time levels t = 0.01 and 0.03 respectively with Δt = 0.00001, Re = 50 and τ = 0.5 for v component

	t = 0.01			t = 0.03		
Mesh Point	**Numerical v**	**Exact v**	**Abs. Error v**	**Numerical v**	**Exact v**	**Abs. Error v**
(0.1, 0.1)	-2.11E-03	-2.11E-03	2.11E-06	-6.33E-03	-6.33E-03	2.11E-06
(0.2, 0.1)	1.03E-01	1.03E-01	2.06E-06	9.91E-02	9.91E-02	1.99E-06
(0.3, 0.1)	2.08E-01	2.08E-01	2.02E-06	2.05E-01	2.05E-01	1.86E-06

Table 18: Comparison of Numerical and Exact solutions at time levels t = 0.1 and 0.3 respectively with Δt = 0.00001, Re = 50 and τ = 0.5 for u component

	t = 0.1			t = 0.3		
Mesh Point	**Numerical u**	**Exact u**	**Abs. Error u**	**Numerical u**	**Exact u**	**Abs. Error u**
(0.1, 0.1)	1.93E-01	1.93E-01	1.24E-06	1.80E-01	1.80E-01	1.25E-07
(0.2, 01)	2.79E-01	2.79E-01	2.92E-06	2.31E-01	2.31E-01	1.03E-06
(0.3, 0.1)	3.65E-01	3.65E-01	4.58E-06	2.82E-01	2.82E-01	1.98E-06

Proof: Let $\phi(x)$ be the cubic Hyperbolic B-spline interpolant of u(x), then using the approximations, we will get the following,

$$|u'(x_i) - U'(x_i)| = |u'(x_i) - \phi'(x_i)| + |\phi'(x_i) - U'(x_i)| \leq |u'(x_i) - \phi'(x_i)| + |\phi'(x_i) - U'(x_i)|$$

Where first term of the aforementioned R.H.S. of the above-mentioned equation has order 4 by [63]. For second term, it is considered that,

$$|\phi'(x_i) - U'(x_i)| = \sum_{k=-1}^{n+1} m_k UAHB'_k(x_i) - \sum_{j=-1}^{n+1} w_{ij}^{(1)} U(x_j)$$

$$|\phi'(x_i) - U'(x_i)| = \sum_{k=-1}^{n+1} m_k \sum_{j=-1}^{n+1} w_{ij}^{(1)} UAHB_k(x_j) - \sum_{j=-1}^{n+1} w_{ij}^{(1)} U(x_j)$$

$$|\phi'(x_i) - U'(x_i)| = \sum_{j=-1}^{n+1} w_{ij}^{(1)} [\sum_{k=-1}^{n+1} m_k UAHB_k(x_j) - U(x_j)]$$

$$|\phi'(x_i) - U'(x_i)| = w_{i1}^{(1)}[\phi(x_1) - U(x_1))] + w_{in}^{(1)}[\phi(x_n - U(x_n)] + \sum_{j=-1, j\neq 1, j\neq n}^{n+1} w_{ij}^{(1)}[\phi(x_j) - U(x_j)]$$

$$|\phi'(x_i) - U'(x_i)| = O(\phi^4) + O(\phi^4) + 0 = O(\phi^4)$$

Table 19: Comparison of Numerical and Exact solutions at time levels t = 0.1 and 0.3 respectively with Δt = 0.00001, Re = 50 and τ = 0.5 for v component

	t = 0.1			t = 0.3		
Mesh Point	**Num. v**	**Exact v**	**Abs. Error v**	**Num. v**	**Exact v**	**Abs. Error v**
(0.1, 0.1)	-0.02148	-0.02148	1.99E-06	-0.07702	-0.07702	2.16E-06
(0.2, 01)	0.085927	0.085929	1.84E-06	0.051346	0.051348	1.94E-06
(0.3, 0.1)	0.193339	0.19334	1.45E-06	0.179717	0.179718	1.03E-06

Table 20: Comparison of Numerical results for u component of present scheme with [15][6][5] where grid size is 20×20 and Δt = 0.0001 at time level t = 0.625, Re = 50

u(x, t)

Mesh Point	**Srivastava et al. I-LFDM [15]**	**Shukla et al. MCB DQM [6]**	**Tamsir et al. [5]**	**Present Method**
(0.1, 0.1)	0.97146	0.97056	0.970558	0.965404
(0.3, 0.1)	1.1528	1.15152	1.15152	1.151265
(0.2, 0.2)	0.86308	0.86244	0.862434	0.851284

Table 21: Comparison of Numerical results for v component of present scheme with [5][15][6] where grid size is 20×20 and Δt = 0.0001 at time level t = 0.625, Re = 50

v(x, t)

MP	**Srivastava et al. I-LFDM [15]**	**Shukla et al. MCBDQM [6]**	**Tamsir et al. [5]**	**Present Method**
(0.1, 0.1)	0.09869	0.09842	0.098419	0.102491
(0.3, 0.1)	0.14158	0.14107	0.14107	0.145001
(0.2, 0.2)	0.16754	0.16732	0.167317	0.173746

Similarly

$$|u''(x_i) - U''(x_i)| = |u'(x_i) - \phi'(x_i)| + |\phi''(x_i) - U''(x_i)|$$

$$|u''(x_i) - U''(x_i)| \leq |u'(x_i) - \phi'(x_i)| + |\phi''(x_i) - U''(x_i)|$$

Where first term of the above-mentioned R.H.S. of the aforementioned equation has order 4 by [63]. For second term, it is considered that,

$$\phi''(x_i) - U''(x_i) = \sum_{k=-1}^{n+1} m_k UAHB''_k(x_i) - \sum_{j=-1}^{n+1} w_{ij}^{(2)} U(x_j)$$

$$\phi''(x_i) - U''(x_i) = \sum_{k=-1}^{n+1} m_k \sum_{j=-1}^{n+1} w_{ij}^{(2)} UAHB_k(x_j) - \sum_{j=-1}^{n+1} w_{ij}^{(2)} U(x_j)$$

$$\phi''(x_i) - U''(x_i) = \sum_{j=-1}^{n=1} w_{ij}^{(2)} [\sum_{k=-1}^{n+1} m_k UAHB_k(x_j) - U(x_j)]$$

Table 22: Comparison of Numerical results for both u and v components at time level t = 0.01 and Re = 100

	MeshPoints	**ETDRK4[59]**		**Method of lines[60]**		**FEM [60]**		**[61]**	**PresentMethod**
		$N_x = 20$	$N_x = 40$	$N_x = 20$	**Nx = 40**	$N_x = 20$	$N_x = 40$	$N_x = 40, \Delta t = .0005$	$N_x = 40, \Delta t = .0005$
u	(0.1, 0.1)	0.07258	0.07253	0.07257	0.07253	0.07257	0.07252	0.072517	0.088207
	(0.2, 0.8)	0.28846	0.28836	0.28846	0.28836	0.28842	0.28835	0.277556	0.300862
	(0.4, 0.4)	0.72206	0.72178	0.72205	0.72178	0.7221	0.72179	0.721622	0.87046
	(0.7, 0.1)	0.20113	0.20106	0.20112	0.20106	0.20113	0.20107	0.20477	0.253024
v	(0.1, 0.1)	0.43302	0.43162	0.43302	0.43173	0.44336	0.43178	0.431192	0.74241
	(0.2, 0.8)	-0.1239	-0.1218	-0.1239	-0.1218	-0.1237	-0.1218	-0.12434	-0.55758
	(0.4, 0.4)	1.65573	1.65336	1.65571	1.65335	1.65499	1.65316	1.65234	2.524097
	(0.7, 0.1)	0.06571	0.06681	0.06571	0.06679	0.06621	0.06692	0.066822	-0.17569

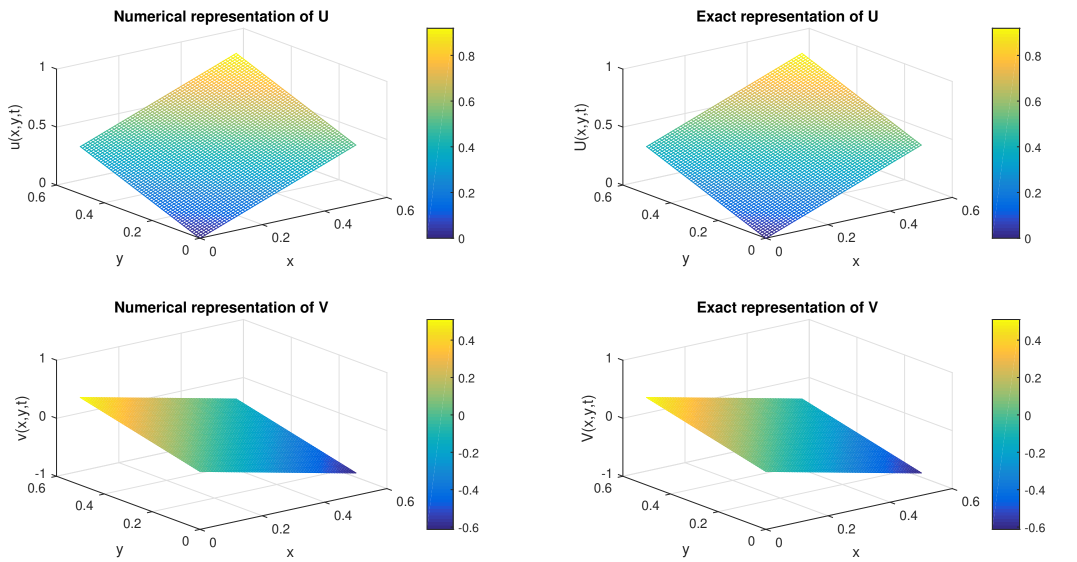

Fig. 5: Graphical presentation of Numerical and Exact solutions of u and v components at time level t = 0.1 with $\Delta t = 10^{-4}$, Re = 200 and $\tau = 1$

$$\phi''(x_i) - U''(x_i) = w_{i1}^{(2)}[g(x_1) - U(x_1)] + w_{in}^{(2)}[g(x_n) - U(x_n)] + \sum_{j=-1, j\neq 1, j\neq n}^{n+1} w_{ij}^{(2)}[g(x_j) - U(x_j)]$$

$$\phi''(x_i) - U''(x_i) = O(\phi^4) + O(\phi^4) + 0 = O(\phi^4)$$

4 Stability

In the present section, stability of the proposed scheme is enlightened using the notion of matrix stability analysis method. By following the concept of stability from [30][65] and then by applying the above-mentioned approximations in equation (1) and (2) respectively, a newly formed system will be fetched as follows:

$$\frac{\partial u}{\partial t} = \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - u\frac{\partial u}{\partial x} - v\frac{\partial u}{\partial y}$$

$$\frac{\partial u_{ij}}{\partial t} = \nu\left[\sum_{k=1}^{N_x} a_{ik}^{(2)} u_{kj} + \sum_{k=1}^{N_y} b_{jk}^{(2)} u_{ik}\right] - u_{ij}\left[\sum_{k=1}^{N_x} a_{ik}^{(1)} u_{kj}\right] - v_{ij}\left[\sum_{k=1}^{N_y} b_{jk}^{(1)} u_{ik}\right]$$

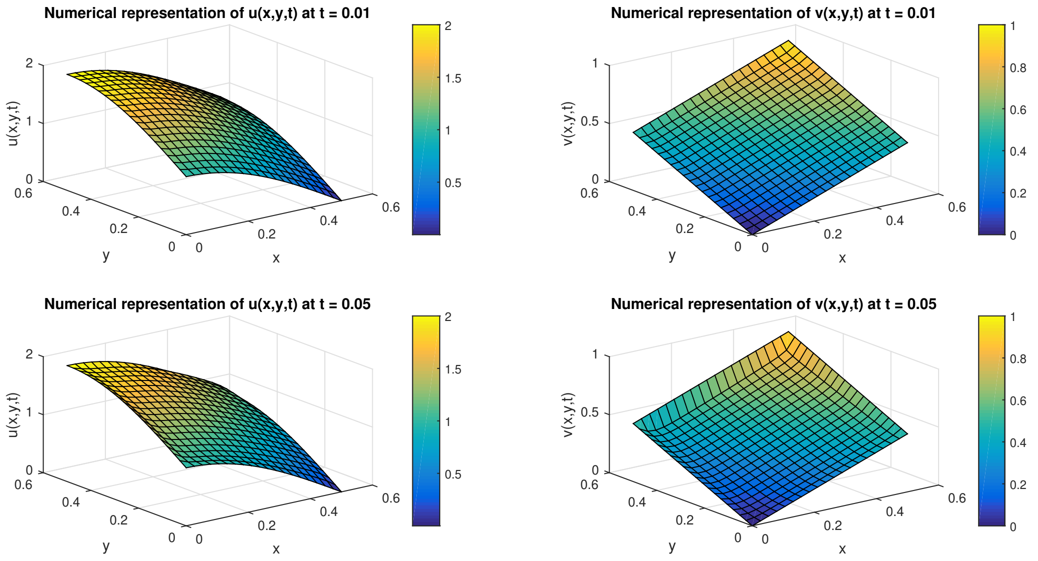

Fig. 6: Graphical representation of Numerical solutions of u and v components at time levels t = 0.01 and 0.05 respectively with Δt = 0.0001, Re = 50, τ = 0.1 and N = 20

Table 23: Comparison of numerical values at different mesh points with [60]

		N_x = 80 and Δt = 0.005, Re = 100			
		Singh and Kumar [61]		Present Method	
Mesh Point		t = 0.5	t = 1.0	t = 0.5	t = 1.0
(0.1, 0.1)	u	0.0151	0.007264	0.01526	0.007387
	v	0.12163	0.055424	0.12285	0.056294
(0.2, 0.8)	u	0.15842	0.080754	0.15836	0.081835
	v	0.98739	0.581761	0.96896	0.587791
(0.4, 0.4)	u	0.12822	0.070449	0.13009	0.071795
	v	0.70021	0.369	0.70256	0.371983

$$\frac{\partial u_{ij}}{\partial t} = \nu[\sum_{k=2}^{N_x-1} a_{ik}^{(2)} u_{kj} + \sum_{k=2}^{N_y-1} b_{jk}^{(2)} u_{ik}] - u_{ij}[\sum_{k=2}^{N_x-1} a_{ik}^{(1)} u_{kj}] - v_{ij}[\sum_{k=2}^{N_y-1} b_{jk}^{(1)} u_{ik}] + F_{ij} \tag{49}$$

$$F_{ij} = \nu[a_{i1}^{(2)} u_{1j} + a_{iN_x}^{(2)} u_{N_x j} + b_{j1}^{(2)} u_{i1} + b_{jN_y}^{(2)} u_{iN_y}] - u_{ij}[a_{i1}^{(1)} u_{1j} + a_{iN_x}^{(1)} u_{jN_x}] - v_{ij}[b_{j1}^{(1)} u_{i1} + b_{jN_y}^{(1)} u_{iN_y}]$$

$$\frac{\partial v}{\partial t} = \nu(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}) - u\frac{\partial v}{\partial x} - v\frac{\partial v}{\partial y}$$

$$\frac{\partial v_{ij}}{\partial t} = \nu[\sum_{k=1}^{N_x} a_{ik}^{(2)} v_{kj} + \sum_{k=1}^{N_y} b_{jk}^{(2)} v_{ik}] - u_{ij}[\sum_{k=1}^{N_x} a_{ik}^{(1)} v_{kj}] - v_{ij}[\sum_{k=1}^{N_y} b_{jk}^{(1)} v_{ik}]$$

$$\frac{\partial v_{ij}}{\partial t} = \nu[\sum_{k=2}^{N_x-1} a_{ik}^{(2)} v_{kj} + \sum_{k=2}^{N_y-1} b_{jk}^{(2)} v_{ik}] - u_{ij}[\sum_{k=2}^{N_x-1} a_{ik}^{(1)} v_{kj}] - v_{ij}[\sum_{k=2}^{N_y-1} b_{jk}^{(1)} v_{ik}] + G_{ij} \tag{50}$$

$$G_{ij} = \nu[a_{i1}^{(2)} v_{1j} + a_{iN_x}^{(2)} v_{N_x j} + b_{j1}^{(2)} v_{i1} + b_{jN_y}^{(2)} v_{iN_y}] - u_{ij}[a_{i1}^{(1)} v_{1j} + a_{iN_x}^{(1)} v_{jN_x}] - v_{ij}[b_{j1}^{(1)} v_{i1} + b_{jN_y}^{(1)} v_{iN_y}]$$

and $A_2 = [a_{ij}^{(2)}], A_1 = [a_{ij}^{(1)}] and B_2 = [b_{ij}^{(2)}], B_1 = [b_{ij}^{(1)}]$ This system formed by equations (49) and (50) can be written as follows:

$$\frac{dU}{dt} = BU + H \tag{51}$$

Where B =

$$\begin{bmatrix} A & O \\ O & A \end{bmatrix}$$

Where

$$A = -u_{ij}A_1 - v_{ij}B_1 + \nu[A_2 + B_2]$$

Stability of the developed scheme depends upon the eigen values of the matrix B given in equation (51). On checking the stability of proposed scheme at different grid points by Matrix stability analysis method, this notion

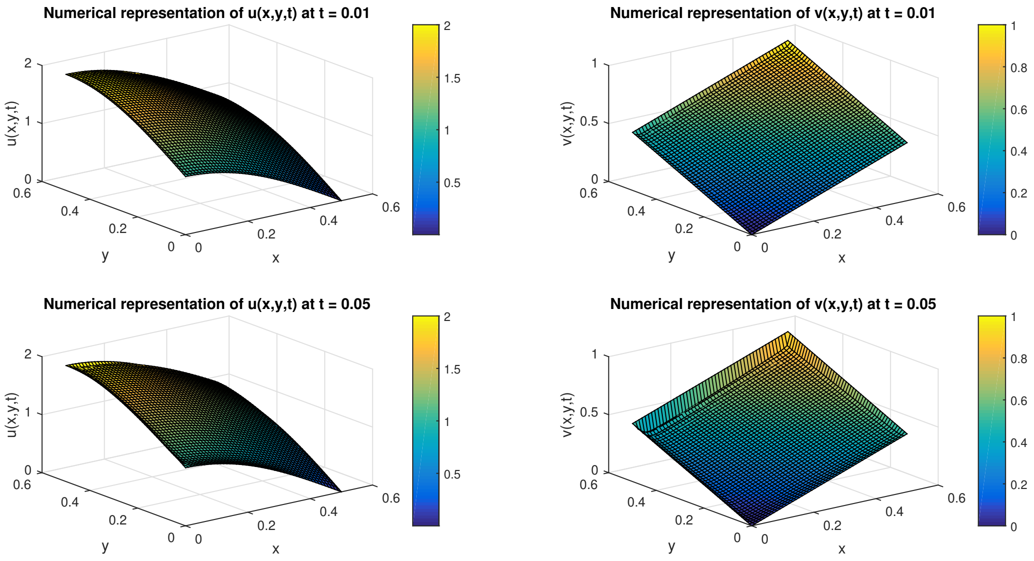

Fig. 7: Graphical representation of Numerical solutions of u and v components at time levels t = 0.01 and 0.05 respectively with Δt = 0.0001, Re = 200, τ = 0.1 and number of grid points = 50

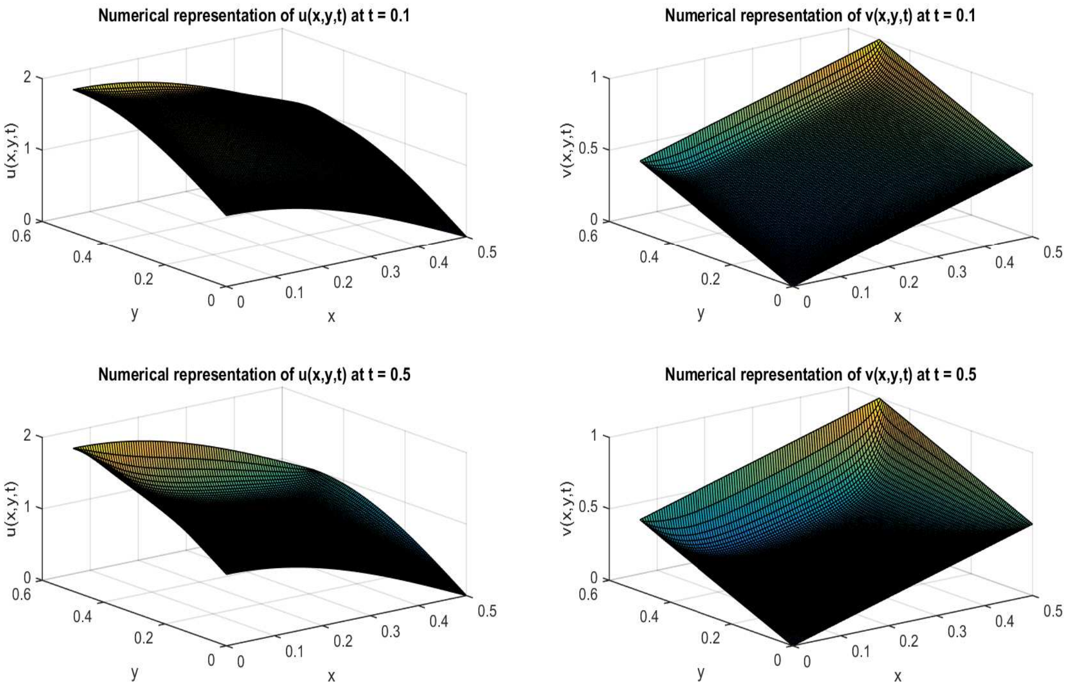

Fig. 8: Graphical representation of Numerical solutions of u and v components at time levels t = 0.1 and 0.5 respectively with Δt = 0.0001, Re = 50, τ = 0.5 and N = 100

Table 24: Errors of u and v components with Re = 500, τ = 2, Δt = 0.00001 for given grid points at time levels t = 0.01 and 0.05 respectively

	t = 0.01		t = 0.01		t = 0.05		t = 0.05	
	u		v		u		v	
n	L_2	L_∞	L_2	L_∞	L_2	L_∞	L_2	L_∞
10	2.45E-03	5.28E-04	2.45E-03	5.28E-04	1.23E-02	2.65E-03	1.23E-02	2.65E-03
30	9.93E-04	1.67E-04	9.93E-04	1.67E-04	4.87E-03	8.43E-04	4.87E-03	8.43E-04
50	6.85E-04	9.95E-05	6.85E-04	9.95E-05	3.29E-03	4.89E-04	3.29E-03	4.89E-04

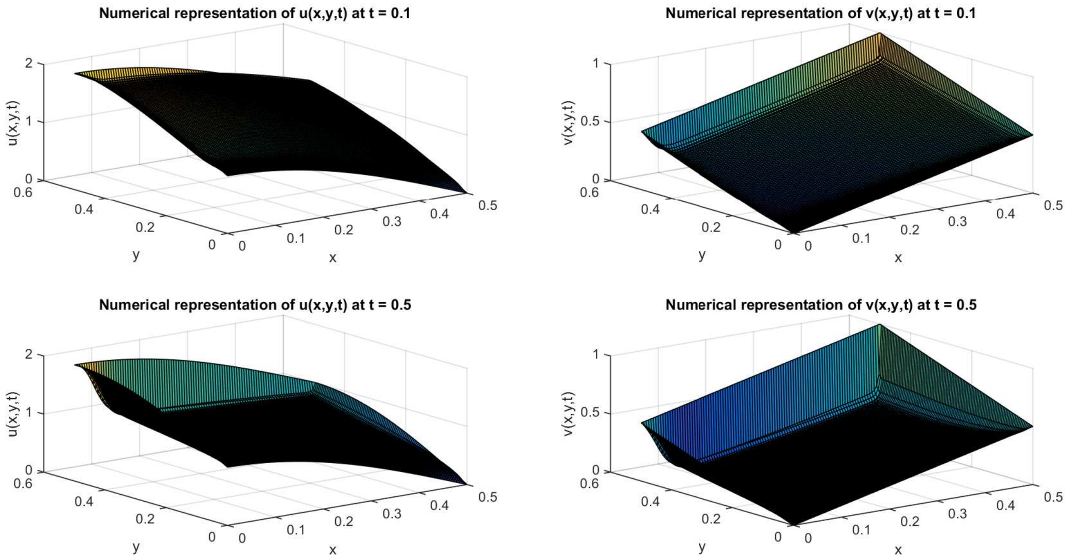

Fig. 9: Graphical representation of Numerical solutions of u and v components at time levels t = 0.1 and 0.5 respectively with Δt = 0.0001, Re = 500, τ = 0.5 and N = 100

Table 25: Order of convergence at time level t = 1 for U component with aid of Example 1

	t = 1			
N	L_2U	ROC	$L_\infty U$	ROC
10	2.82E-03	-	4.45E-03	-
20	7.15E-04	1.9807	1.00E-03	2.1477
40	1.95E-04	1.8756	2.06E-04	2.2865

Table 26: Order of convergence at time level t = 2 for U component with aid of Example 1

	t = 2			
N	L_2U	ROC	$L_\infty U$	ROC
10	2.13E-03	-	4.06E-03	-
20	3.58E-04	2.573	1.00E-03	2.0144
40	9.18E-05	1.9619	2.07E-04	2.2792

came into light that proposed scheme is unconditionally stable. Stability of the proposed method is discussed by means of following Figure 13.

5 Conclusion

Because of, enormous applications of coupled 2D Burgers' equation in different areas of sciences and engineering, attention of researchers got triggered towards its numerical solution. In most of the non-linear partial differential equations, analytical solution is not an easy task to fetch. Thus, finding the numerical solution of such partial differential equations became the necessity of time. For this purpose a range of numerical schemes have been developed by many researchers. This paper addressed the numerical solution of such non-linear partial differential equations by implementing uniform

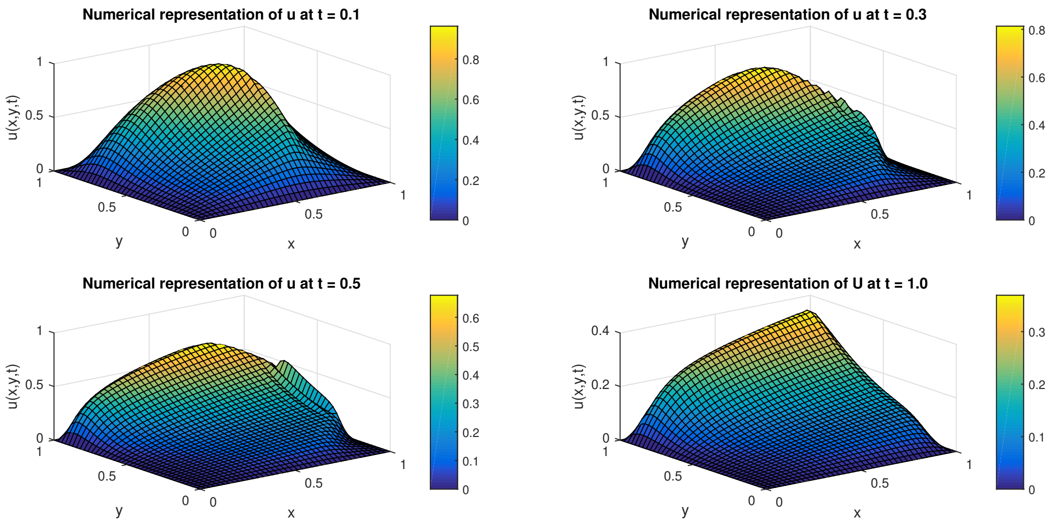

Fig. 10: Graphical representation of Numerical u with N_x= 40, Δt = 0.005 at different time levels with Re = 100

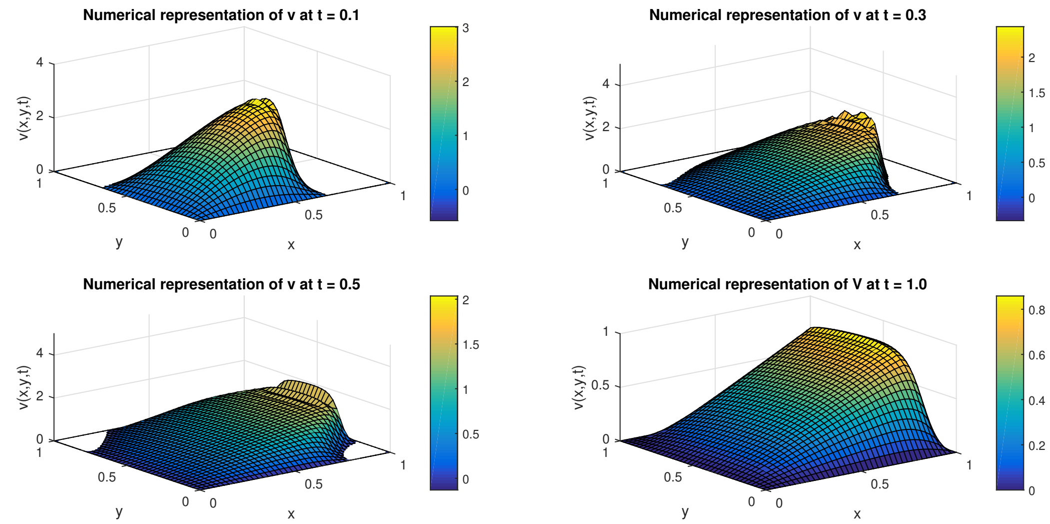

Fig. 11: Graphical representation of Numerical v with N_x = 40, Δt = 0.005 at different time levels with Re = 100

Table 27: Order of convergence at time level t = 1 for V component with aid of Example 1

	t = 1			
N	L_2V	**ROC**	$L_\infty V$	**ROC**
10	2.82E-03	-	4.45E-03	-
20	7.15E-04	1.9807	1.00E-03	2.1477
40	1.95E-04	1.8756	2.06E-04	2.2865

Table 28: Order of convergence at time level t = 2 for V component with aid of Example 1

	t = 2			
N	L_2V	**ROC**	$L_\infty V$	**ROC**
10	2.13E-03	-	4.06E-03	-
20	3.58E-04	2.573	1.00E-03	2.0144
40	9.18E-05	1.9619	2.07E-04	2.2792

algebraic hyperbolic (UAH) tension B-spline based DQM. The obtained numerical results were compared

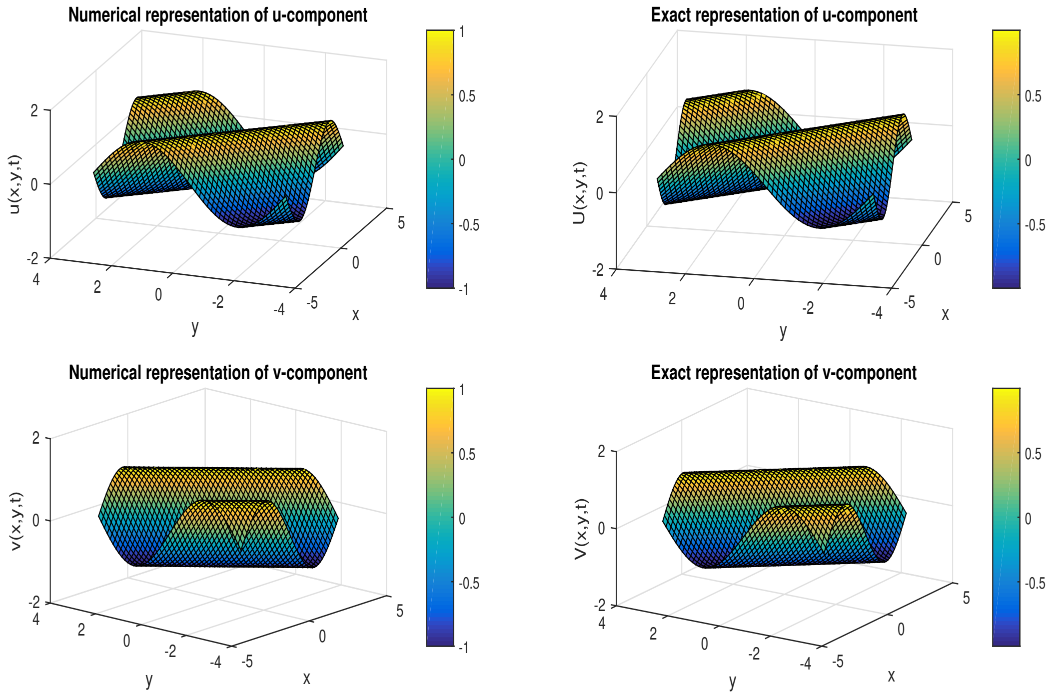

Fig. 12: Graphical representation of Numerical and Exact solutions of u and v components for Re = 100, τ = 2, N = 50 , Δt = 0.00001 at time level t = 0.05

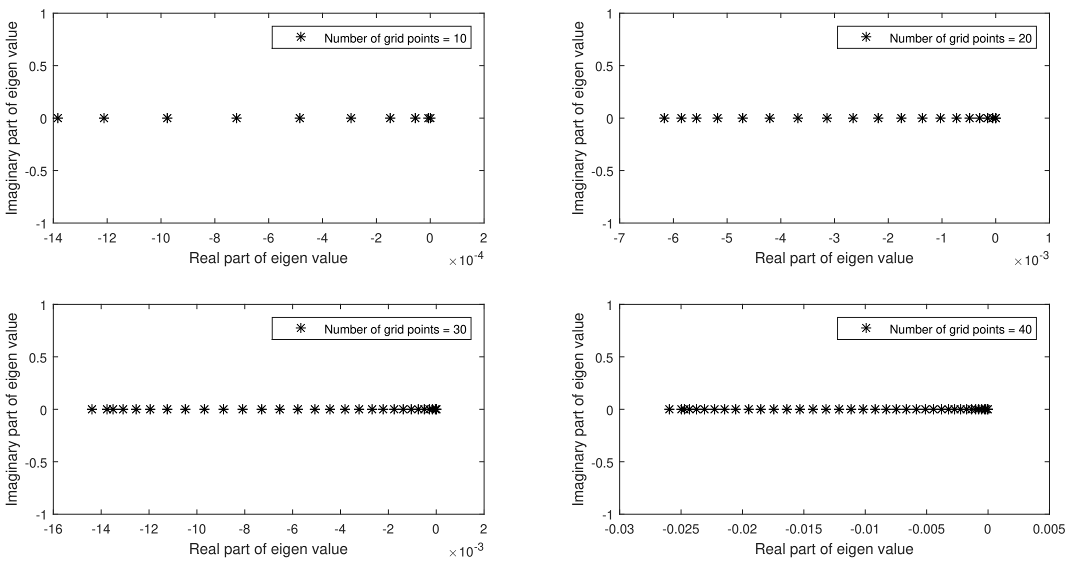

Fig. 13: Stability of proposed scheme for Example 1

Table 29: Comparison of RMS error norm and Relative Error for different number of grid points at time level t = 2 (considered for Example 1)

	RMS error norms U	C.P.U. Time (seconds)	Relative Error U	C.P.U. Time (seconds)
N = 10	3.19E-04	4.43062	3.19E-04	4.43062
N = 20	7.56E-05	13.7068	7.56E-05	13.7074
N = 30	2.64E-05	27.1879	2.64E-05	27.1885
	RMS error norms V	**C.P.U. Time (seconds)**	**Relative Error V**	**C.P.U. Time (seconds)**
N = 10	1.79E-04	4.43062	1.79E-04	4.4312
N = 20	4.25E-05	13.7068	4.25E-05	13.7074
N = 30	1.51E-05	27.1879	1.51E-05	27.1885

Table 30: Comparison of RMS error norm and Relative Error for different number of grid points at time level t = 0.05 (considered for Example 5)

	RMS error norms U	C.P.U. Time (seconds)	Relative Error U	C.P.U. Time (seconds)
N = 10	1.34E-03	1.24765	1.34E-03	1.24842
N = 20	1.13E-03	3.38583	1.13E-03	3.38637
N = 30	9.66E-04	7.55142	9.66E-04	7.55202
	RMS error norms V	**C.P.U. Time (seconds)**	**Relative Error V**	**C.P.U. Time (seconds)**
N = 10	1.34E-03	1.24765	1.34E-03	1.24842
N = 20	1.13E-03	3.38583	1.13E-03	3.38637
N = 30	9.66E-04	7.55142	9.66E-04	7.55202

with the previous ones and with exact solution through L_2 and L_∞ error norms. A good match between the present approximated results with exact solutions asserted that the developed scheme has worth to get employed to solve the non-linear partial differential equations numerically. Stability is an another step to prove that the developed scheme is unconditionally stable for different number of grid points. Hence, it can be confirmed that the proposed method is one of the effective methods to compute the numerical solution of a range of non-linear partial differential equations.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

[1] H. Bateman, “Some recent researches on the motion of fluids,” *Monthly Weather Review*, vol. 43, no. 4, pp. 163–170, 1915.
[2] J. M. Burgers, “A mathematical model illustrating the theory of turbulence,” in *Advances in applied mechanics*, vol. 1, pp. 171–199, Elsevier, 1948.
[3] C. A. Fletcher, “Generating exact solutions of the two-dimensional Burgers’ equations,” *IJNMF*, vol. 3, pp. 213–216, 1983.
[4] M. Tamsir, N. Dhiman, and V. K. Srivastava, “Extended modified cubic B-spline algorithm for nonlinear Burgers’ equation,” *Beni-Suef University journal of basic and applied sciences*, vol. 5, no. 3, pp. 244–254, 2016.
[5] M. Tamsir, V. K. Srivastava, and R. Jiwari, “An algorithm based on exponential modified cubic B-spline differential quadrature method for nonlinear Burgers’ equation,” *Applied Mathematics and Computation*, vol. 290, pp. 111–124, 2016.
[6] H. Shukla, M. Tamsir, V. K. Srivastava, and J. Kumar, “Numerical solution of two dimensional coupled viscous Burger equation using modified cubic B-spline differential quadrature method,” *AIP advances*, vol. 4, no. 11, p. 117134, 2014.
[7] R. Abazari and A. Borhanifar, “Numerical study of the solution of the Burgers and coupled Burgers equations by a differential transformation method,” *Computers & Mathematics with Applications*, vol. 59, no. 8, pp. 2711–2722, 2010.
[8] R. Mittal and R. Jiwari, “Differential quadrature method for two-dimensional Burgers’ equations,” *International Journal for Computational Methods in Engineering Science and Mechanics*, vol. 10, no. 6, pp. 450–459, 2009.
[9] A. R. Bahadır, “A fully implicit finite-difference scheme for two-dimensional Burgers’ equations,” *Applied Mathematics*

and Computation, vol. 137, no. 1, pp. 131–137, 2003.
[10] P. Jain and D. Holla, "Numerical solutions of coupled Burgers' equation," *International Journal of Non-Linear Mechanics*, vol. 13, no. 4, pp. 213–222, 1978.
[11] H. Zhu, H. Shu, and M. Ding, "Numerical solutions of two-dimensional Burgers' equations by discrete adomian decomposition method," *Computers & Mathematics with Applications*, vol. 60, no. 3, pp. 840–848, 2010.
[12] C.-M. Fan, C.-H. Yang, and M.-H. Gu, "Applications of the local rbf collocation method and the fictitious time integration method for Burgers' equations," *Procedia Engineering*, vol. 79, pp. 569–574, 2014.
[13] M. Khan, "A novel solution technique for two dimensional Burger's equation," *Alexandria Engineering Journal*, vol. 53, no. 2, pp. 485–490, 2014.
[14] E. Hızel and S. Küçükarslan, "Homotopy perturbation method for (2+ 1)-dimensional coupled Burgers system," *Nonlinear Analysis: Real World Applications*, vol. 10, no. 3, pp. 1932–1938, 2009.
[15] V. K. Srivastava, M. K. Awasthi, and S. Singh, "An implicit logarithmic finite-difference technique for two dimensional coupled viscous Burgers' equation," *Aip Advances*, vol. 3, no. 12, p. 122105, 2013.
[16] V. Srivastava and M. Tamsir, "Crank-nicolson semi implicit approach for numerical solutions of two-dimensional coupled nonlinear Burgers' equations," *International Journal of Applied Mechanics andEngineering*, vol. 17, no. 2, p. 571, 2012.
[17] T. Zhanlav, O. Chuluunbaatar, and V. Ulziibayar, "Higher-order accurate numerical solution of unsteady Burgers' equation," *Applied Mathematics and Computation*, vol. 250, pp. 701–707, 2015.
[18] R. Bellman, B. Kashef, and J. Casti, "Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations," *Journal of computational physics*, vol. 10, no. 1, pp. 40–52, 1972.
[19] J. Quan and C. Chang, "New insights in solving distributed system equations by the quadrature method—i. analysis," *Computers & Chemical Engineering*, vol. 13, no. 7, pp. 779–788, 1989.
[20] J. Quan and C.-T. Chang, "New insights in solving distributed system equations by the quadrature method—ii. numerical experiments," *Computers & Chemical Engineering*, vol. 13, no. 9, pp. 1017–1024, 1989.
[21] C. Shu and B. Richards, "High resolution of natural convection in a square cavity by generalized differential quadrature," in *Proceedings of the 3rd international conference on advances in numeric methods in engineering: theory and application, Swansea, UK*, pp. 978–985, 1990.
[22] C. Shu, *Generalized differential-integral quadrature and application to the simulation of incompressible viscous flows including parallel computation*. PhD thesis, ProQuest Dissertations & Theses,, 1991.
[23] Ö. Civalek, "Application of differential quadrature (dq) and harmonic differential quadrature (hdq) for buckling analysis of thin isotropic plates and elastic columns," *Engineering Structures*, vol. 26, no. 2, pp. 171–186, 2004.
[24] H. Zhong, "Spline-based differential quadrature for fourth order differential equations and its application to kirchhoff plates," *Applied Mathematical Modelling*, vol. 28, no. 4, pp. 353–366, 2004.
[25] H. Zhong and M. Lan, "Solution of nonlinear initial-value problems by the spline-based differential quadrature method," *Journal of Sound and Vibration*, vol. 296, no. 4-5, pp. 908–918, 2006.
[26] A. Korkmaz and İ. Dağ, "Shock wave simulations using sinc differential quadrature method," *Engineering Computations*, 2011.
[27] A. Korkmaz and I. Dag, "Polynomial based differential quadrature method for numerical solution of nonlinear Burgers' equation," *Journal of the Franklin Institute*, vol. 348, no. 10, pp. 2863–2875, 2011.
[28] A. Korkmaz, A. M. Aksoy, and I. Dag, "Quartic B-spline differential quadrature method," *Int. J. Nonlinear Sci*, vol. 11, no. 4, pp. 403–411, 2011.
[29] G. Arora and B. K. Singh, "Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method," *Applied Mathematics and Computation*, vol. 224, pp. 166–177, 2013.
[30] G. Arora and V. Joshi, "A computational approach using modified trigonometric cubic B-spline for numerical solution of Burgers' equation in one and two dimensions," *Alexandria Engineering Journal*, vol. 57, no. 2, pp. 1087–1098, 2018.
[31] R. Mittal and S. Dahiya, "Numerical simulation of three-dimensional telegraphic equation using cubic B-spline differential quadrature method," *Applied Mathematics and Computation*, vol. 313, pp. 442–452, 2017.
[32] R. Mittal and R. Jiwari, "A differential quadrature method for numerical solutions of Burgers'-type equations," *International Journal of Numerical Methods for Heat & Fluid Flow*, 2012.
[33] R. Jiwari, R. Mittal, and K. K. Sharma, "A numerical scheme based on weighted average differential quadrature method for the numerical solution of Burgers' equation," *Applied Mathematics and Computation*, vol. 219, no. 12, pp. 6680–6691, 2013.
[34] H. Shukla, M. Tamsir, R. Jiwari, and V. K. Srivastava, "A numerical algorithm for computation modelling of 3d nonlinear wave equations based on exponential modified cubic B-spline differential quadrature method," *International Journal of Computer Mathematics*, vol. 95, no. 4, pp. 752–766, 2018.
[35] R. Jiwari, S. Pandit, and R. Mittal, "Numerical simulation of two-dimensional sine-gordon solitons by differential quadrature method," *Computer Physics Communications*, vol. 183, no. 3, pp. 600–616, 2012.
[36] R. Mittal and R. Jiwari, "Numerical solution of two-dimensional reaction–diffusion brusselator system," *Applied mathematics and computation*, vol. 217, no. 12, pp. 5404–5415, 2011.
[37] A. Korkmaz, "Numerical algorithms for solutions of Korteweg-de Vries equation," *Numerical methods for partial differential equations*, vol. 26, no. 6, pp. 1504–1521, 2010.
[38] S. Tomasiello, "Differential quadrature method: application to initial-boundary-value problems," *Journal of Sound and Vibration*, vol. 218, no. 4, pp. 573–585, 1998.
[39] S. Tomasiello, "Numerical stability of dq solutions of wave problems," *Numerical Algorithms*, vol. 57, no. 3, pp. 289–312, 2011.

[40] A. Al Kaisy, R. A. Esmaeel, and M. M. Nassar, "Application of the differential quadrature method to the longitudinal vibration of non-uniform rods," *Engineering Mechanics*, vol. 14, no. 5, pp. 303–310, 2007.
[41] E. Mainar, J. M. Peña, and J. Sánchez-Reyes, "Shape preserving alternatives to the rational Bézier model," *Computer aided geometric design*, vol. 18, no. 1, pp. 37–60, 2001.
[42] J. Zhang, "C-curves: an extension of cubic curves," *Computer Aided Geometric Design*, vol. 13, no. 3, pp. 199–217, 1996.
[43] J. Zhang, "Two different forms of CB-splines," *Computer aided geometric design*, vol. 14, no. 1, pp. 31–41, 1997.
[44] P. E. Koch and T. Lyche, "Construction of exponential tension B-splines of arbitrary order," in *Curves and surfaces*, pp. 255–258, Elsevier, 1991.
[45] Y. Lü, G. Wang, and X. Yang, "Uniform hyperbolic polynomial B-spline curves," *Computer Aided Geometric Design*, vol. 19, no. 6, pp. 379–393, 2002.
[46] E. Mainar and J. M. Peña, "A basis of C-Bézier splines with optimal properties," *Computer Aided Geometric Design*, vol. 19, no. 4, pp. 291–295, 2002.
[47] Q. Chen and G. Wang, "A class of Bézier-like curves," *Computer Aided Geometric Design*, vol. 20, no. 1, pp. 29–39, 2003.
[48] G. Wang, Q. Chen, and M. Zhou, "NUAT B-spline curves," *Computer Aided Geometric Design*, vol. 21, no. 2, pp. 193–205, 2004.
[49] M. K. Jena, P. Shunmugaraj, and P. Das, "A subdivision algorithm for trigonometric spline curves," *Computer Aided Geometric Design*, vol. 19, no. 1, pp. 71–88, 2002.
[50] M. K. Jena, P. Shunmugaraj, and P. Das, "A non-stationary subdivision scheme for generalizing trigonometric spline surfaces to arbitrary meshes," *Computer Aided Geometric Design*, vol. 20, no. 2, pp. 61–77, 2003.
[51] L. Ya-Juan and W. Guo-Zhao, "Two kinds of B-basis of the algebraic hyperbolic space," *Journal of Zhejiang University-SCIENCE A*, vol. 6, no. 7, pp. 750–759, 2005.
[52] J. Zhang, F.-L. Krause, and H. Zhang, "Unifying C-curves and H-curves by extending the calculation to complex numbers," *Computer Aided Geometric Design*, vol. 22, no. 9, pp. 865–883, 2005.
[53] J. Zhang and F.-L. Krause, "Extending cubic uniform B-splines by unified trigonometric and hyperbolic basis," *Graphical Models*, vol. 67, no. 2, pp. 100–119, 2005.
[54] G. Wang *et al.*, "Unified and extended form of three types of splines," *Journal of computational and applied mathematics*, vol. 216, no. 2, pp. 498–508, 2008.
[55] G. Xu and G.-Z. Wang, "AHT bézier curves and NUAHT B-spline curves," *Journal of Computer Science and Technology*, vol. 22, no. 4, pp. 597–607, 2007.
[56] N. Alinia and M. Zarebnia, "A new tension B-spline method for third-order self-adjoint singularly perturbed boundary value problems," *Journal of Computational and Applied Mathematics*, vol. 342, pp. 521–533, 2018.
[57] N. Alinia and M. Zarebnia, "A numerical algorithm based on a new kind of tension B-spline function for solving Burgers-Huxley equation," *Numerical Algorithms*, vol. 82, no. 4, pp. 1121–1142, 2019.
[58] C. Shu, *Differential quadrature and its application in engineering*. Springer Science & Business Media, 2012.
[59] H. P. Bhatt and A.-Q. M. Khaliq, "Fourth-order compact schemes for the numerical simulation of coupled Burgers' equation," *Computer Physics Communications*, vol. 200, pp. 117–138, 2016.
[60] P. Arminjon and C. Beauchamp, "Numerical solution of Burgers' equations in two space dimensions," *Computer Methods in Applied Mechanics and Engineering*, vol. 19, no. 3, pp. 351–365, 1979.
[61] B. K. Singh and P. Kumar, "A novel approach for numerical computation of Burgers' equation in (1+1) and (2+1) dimensions," *Alexandria Engineering Journal*, vol. 55, no. 4, pp. 3331–3344, 2016.
[62] J. Biazar and H. Aminikhah, "Exact and numerical solutions for non-linear Burger's equation by VIM," *Mathematical and Computer Modelling*, vol. 49, no. 7-8, pp. 1394–1400, 2009.
[63] M. Ghasemi, "High order approximations using spline-based differential quadrature method: Implementation to the multi-dimensional pdes," *Applied Mathematical Modelling*, vol. 46, pp. 63–80, 2017.
[64] J. W. Daniel and B. K. Swartz, "Extrapolated collocation for two-point boundary-value problems using cubic splines," *IMA Journal of Applied Mathematics*, vol. 16, no. 2, pp. 161–174, 1975.
[65] M. Jain, *Numerical solution of differential equations*. Wiley, 1983.

Varun Joshi is working as an Assistant Professor in the "Department of Mathematics, Lovely Professional University, Punjab, India". Dr. Varun has obtained the Ph.D. degree in Mathematics from Lovely Professional University. His area of research is related to numerical analysis, partial differential equations etc. He has a number of publications in the journals in international repute.

Mamta Kapoor is working as an Assistant Professor in the "Department of Mathematics, Lovely Professional University, Punjab, India". Her area of research is numerical analysis, study of partial differential equations, analytical solutions of differential equations etc.