Research Article

A novel technique for numerical approximation of 1D non-linear coupled Burgers' equation by using cubic Hyperbolic B-spline based Differential quadrature method

Mamta Kapoor[1], Varun Joshi[2]
[1, 2]Department of Mathematics,Lovely Professional University, Punjab, India - 144411.
Email id:*mamtakapoor.78@yahoo.com*

Abstract
In this paper, a novel scheme, cubic Hyperbolic B-spline-based Differential Quadrature Method, is proposed for the solution of 1D non-linear viscous coupled Burgers' equation. The numerical approximation of this mentioned equation is obtained by using the Hyperbolic B-spline-based Differential quadrature method. Hyperbolic B-spline is used as a basis function in DQM in order to obtain weighting coefficients, then received a set of ODEs is solved by using Strong Stability Preserving Runge Kutta-43 scheme(SSP-RK43 scheme). The accuracy and effectiveness of this proposed scheme are tested by using three examples. The obtained results are matched with the previous results present in the literature of other methods as well as with the exact solutions by means of L_2 and L_∞Error norms mainly, in the form of tables and figures, the proposed scheme has produced better results. The analysis of the stability of this proposed scheme is also discussed by means of examples at different grid points, which indicated that the proposed scheme cubic Hyperbolic B-spline-based DQM is unconditionally stable. For the stability of the present scheme, the matrix stability analysis method is used.
Keywords: *Hyperbolic B-spline;DQM; SSP-RK43 scheme;* L_2 *and* L_∞ *Error norms;matrix stability analysis method.*

1. Introduction

Coupled1D viscous Burgers' equation is one of the finest and the simplest models of sedimentation and evolution of the uplifted volume concentration, having two kinds of particles in fluid suspensions and colloids under the gravity effect. Coupled 1D Burgers' equation was derived by Esipov[1] in (1995) for the study of a model of Poly dispersive sedimentation. Governing equations of Coupled 1-D Burgers' equations are given as follows from [9].

$$u_t + \delta u_{xx} + \eta u u_x + \alpha(uv)_x = 0 \quad \textbf{(1.1)}$$

$$v_t + \mu v_{xx} + \xi v v_x + \beta(uv)_x = 0 \quad \textbf{(1.2)}$$

Where initial conditions are given as follows

$$\begin{cases} u(\mathrm{x},0) = f_1(\mathrm{x}) \\ v(\mathrm{x},0) = f_2(\mathrm{x}) \end{cases} \quad \text{where } x \in D \quad \textbf{(1.3)}$$

And Dirichlet Boundary conditions are given as follows,

$$\begin{cases} u(\mathrm{x},\mathrm{t}) = g_1(\mathrm{x},\mathrm{t}) \\ v(\mathrm{x},\mathrm{t}) = g_2(\mathrm{x},\mathrm{t}) \end{cases} \quad \text{where } x \in D,\ t > 0 \quad \textbf{(1.4)}$$

$D = \{x: a \le x \le b\}$ is given computational domain. $\delta, \eta, \mu\ and\ \xi$ are the real constants, $\alpha\ and\ \beta$ are the arbitrary constants which depends upon the parameters given in system like Peclet number, the stokes velocity of particles because of gravity and Brownian diffusivity explained by Nee and Duan [46].

$u(\mathrm{x},\mathrm{t})$ and $v(\mathrm{x},\mathrm{t})$are known as the components of velocity, which are to be determined f_1 , f_2, g_1 and g_2 are given functions.$u_t$is given as unsteady term, uu_x is the non-linear convection term, $u_{xx}$is given as diffusion term.

$$u_t=\frac{\partial u}{\partial t}, u_x = \frac{\partial u}{\partial x}, u_y = \frac{\partial u}{\partial y}, u_{xx} = \frac{\partial^2 u}{\partial x^2}, u_{yy} = \frac{\partial^2 u}{\partial y^2}$$

Coupled 1D Burgers' equation is a matter of worth from the numerical aspect because primarily analytical solutions are not obtainable for such equations. Kaya [2]obtained the exact solution of 1D coupled Burgers' equation by the implementation of the Adomian Decomposition Method. Soliman [3]implemented a modified but extended form of the tanh function technique. Several researchers have contributed in order to find the numerical approximations of Coupled 1D Burgers equation. Esipov [1]gave numerical simulations as well as compared the data. The variational iteration method was used to solve the 1D Burgers' equation and coupled Burgers' equation by Abdou and Soliman[4]. The applied conjugate filter approach was applied by Wei and Gu [5]. Chebyshev spectral collocation method was used by Khater et al.[6]. Using the Adomian-Pade technique,Dehghan et al.[7]obtained numerical results of coupled viscous Burgers' equation. Fourier Pseudo-spectral method was applied by Rashid and Ismail[8]. In (2013), Srivastava et al. [9] employed a full implicit Finite Difference scheme to obtain a 1D coupled non-linear Burgers' equation. 1D coupled Burgers' equation is solved by implicit logarithmic FD method [I-LFDM] by Srivastava et al.[10]in this paper numerical method provided a system of non-linear difference equation which was linearized with the help of Newton's methodand obtained system was solved by employing Gauss Elimination Method along with Partial pivoting. Mittal and Jiwari [11]used the concept of Polynomial DQM to attain the solution of 1D non-linear Burgers' equation and 1D non-linear coupled Burgers' equation as well as 2D non-linear Burgers' equation and obtained system of ODE was solved by using RK fourth-order method. Srivastava et al. [12]employed an implicit FD technique to solve 1D coupled Burgers' equation for the uniform grid; in this paper,the Crank-Nicolson scheme formed a system of non-linear difference equation solved at every iteration. The obtained non-linear system was linearized by Newton's iteration method and obtained linear system was solved by Gauss Elimination Method. Lai and Ma [13] proposed a new Lattice Boltzman Model to solve the 1D non-linear viscous coupled Burgers' equation; in this paper, Chapman-Enskog expansion was employed. Mittal and Arora [14]solved 1D coupled viscous Burgers' equation by using the technique of cubic B-spline built Collocation method for uniform mesh point; in this paper, Crank-Nicolson scheme was implemented for time integration, and cubic B-spline scheme was implemented for space integration and stability was also discussed by using Von-Neumann method. Mokhtari et al.[15]employed the Generalized DQM to solve 1D Burgers' equation, 2-D Burgers' equation, as well as 1D, coupled Burgers' equation numerically. In this paper, Polynomial DQM is implemented, and obtained system of ODE is solved by using (TVD – RK method) Total variation diminishing Runge-Kutta method. Salih et al.[16] presented a cubic trigonometric B-spline technique to obtain the numerical approximation of 1D coupled viscous Burgers' equation; in this paper,FD scheme was used to discretize time derivative, and cubic trigonometric B-spline basis function was used as interpolation function in a spatial dimension. The stability ofthe method was checked by the implementation of the Vonn-Neumann method. Raslan et al.[17]gave a numerical technique to approximate the solution of 1D coupled Burgers' equation numerically by implementing the quantic B-spline built Collocation method. The stability of the method was also checked by the Vonn-Neumann method. Liu et al.[18], (2018),employed the Collocation method based upon the Barycentric interpolation function to approximate the solution of coupled 1D viscous Burgers' equation numerically. Coupled viscous Burgers' equation was solved by Li et al.[19]by implementing the technique a New Lattice Boltzmann model. Bhatt and Khaliq[20]gave the method of compact schemes with fourth-order to get the numerical approximation of coupled Burgers' equation.

The basic notion of DQM is taken by the idea of integral quadrature, and DQM was firstly familiarized by Bellman et al. in 1972 [21]. After that, the above-mentioned method of finding weighting coefficients was enhanced by Quan and Chang in 1989 [22, 23]. A significant headway to find the weighing coefficients were given by Shu and Richards in 1990 [24]. The above-mentioned methods to determine weighting coefficients are generalized under the exploration of higher-order approximation by Shu in 1991 [25]. So far, different test functions, like Lagrange polynomials, Legendre polynomial, and B-spline basis functions, have been used to generate different types of DQMs. A comparison was made between DQM and Harmonic DQM for buckling analysis of thin isotropic plates and elastic columns by Civalik [26] in 2004. A Quintic B-spline-based DQM to solve the fourth-order differential equation was given by Zhong [27] in 2004. In comparison, Zhong and Lan [28] proposed spline-based DQM to solve the non-linear initial value problems. Exponential modified cubic B-spline functions are used as test functions in DQM by Tamsir et al. [29]. DQM based on Sinc functions is used by Korkmaz and Dag [30]. Korkmaz and Dag used Polynomial DQM to solve non-linear Burger's Equation [31]. Quartic B-spline-based DQM is introduced to get the weighting coefficients by Korkmaz et al. [32]. Arora and Singh [33] represented modified cubic B-spline-based DQM (MCB-DQM) to obtain a numerical solution of Burgers' equation. Arora and Joshi [34] solved one and two-dimensional non-linear Burger's equation by using modified trigonometric cubic B-spline-based DQM. Whereas Mittal and Dahiya [35] used modified cubic-based B-spline as a basis function in DQM to obtain the numerical solution of 3-dimensional hyperbolic equations in 2017. Mittal and Jiwari [36] used Polynomial DQM to get the numerical solution of the non-linear Burger type equation in 2012. Jiwari et al. [37] used weighted average DQM for the solution of time-dependent Burger's equation with given initial and boundary conditions. In comparison, Shukla et al.[38] proposed an Exponential modified cubic B-spline-based DQM (Expo-MCB-DQM) for the solution of a 3-dimensional non-linear wave equation. A numerical method based upon polynomial DQM to find the numerical solution of the sine-Gordon equation is presented by Jiwari et al. [39]. A numerical study using DQM of a 2-dimensional reaction-diffusion brusselator system is also given by Mittal and Jiwari [40]. 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 like [41], [42], [43], [44], [45].

Hyperbolic B-spline. In previous years, various new splines have been defined and used in the concept of geometrical modeling in "CAGD". For instance, non-uniform algebraic trigonometric (NUAT) B-spline used in [47], an orthogonal basis similar to Legendre basis are used in concept of algebraic-trigonometric polynomial space by Huang, and Wang [48], an orthogonal basis for "NUAT" spline space in [49],2π periodic trigonometric was used by Nouisser et al. [50], normalized spherical B-spline was used by Maes and Bultheel [51]. In the present section, we will introduce the formula of "Hyperbolic B-spline" [54]$HB_i^k$of order k, which is associated with the partition Xdefined by,

$$HB_i^1(x) = \begin{cases} 1 \quad , \quad when\ x_i \le x < x_{i+1} \\ 0,\ otherwise \end{cases} \quad \textbf{(1.5)}$$

and for $k > 1$,

$$HB_i^k(x) = \frac{s(x - x_i)}{s(x_{k+i-1} - x_i)} HB_i^{k-1}(x) + \frac{s(x_{i+k} - x)}{s(x_{i+k} - x_{i+1})} HB_{i+1}^{k-1}(x) \textbf{(1.6)}$$

where $s(x) = sinh(x)$, {sine hyperbolic function of x}

Above both equations satisfy the given properties,

(P1): For k $\geq 2, HB_i^k \in C^{k-2}(x)$

(P2): $HB_i^k(x)$ is a piecewise hyperbolic function.

(P3): $HB_i^k(x) \geq 0$

(P4): Support of $HB_i^k(x) = [x_i,\ x_{i+k}]$

(P5): $HB_i^k \in \ulcorner_k$

Where,

$$\ulcorner_k = \begin{cases} span\left\{\{\sinh(2lx), \cosh(2lx)\}_{l=1}^{\left[\frac{k-1}{2}\right]} \cup \{1\}\right\},\ where\ k\ is\ odd \\ span\left\{\{\sinh((2l-1)x), \cosh((2l-1)x)\}_{l=1}^{\left[\frac{k}{2}\right]}\right\},\ where\ k\ is\ even \end{cases} \quad \textbf{(1.7)}$$

which is known as the space of the hyperbolic polynomial of order k.

The main idea of this paper is to propose a new technique, modified cubic Hyperbolic B-spline DQM, to get the numerical approximation of one-dimensional coupled non-linear Burgers' equation. In the present method, a modified cubic Hyperbolic B-spline has been implemented as the test function in DQM to obtain the values of weighting coefficients. By which the non-linear Burgers' equation will get transformed into the system of the first-order ODE. Later on, the obtained system of equations will get solved by employing the four stages and order 3 SSP-RK43 method. The efficiency and compatibility of this projected method will be confirmed by taking some test problems. Error analysis will be done with the help of discrete and root mean square norms. With the help of threeexamples, the applicability and exactness will be checked. In Section 2, a new technique of DQM is developed by using a modified Hyperbolic B-spline. In Section 2.2, a detailed discussion is given to determine weighting coefficients at different grid points. In Section 3, a detailed discussion is given with the help of examples. With the help of these examples, a comparison is given for the numerical and exact solutions, shown by different tables and figures. In Section 4, a stability analysis of the projected scheme is presented for examples for various grid points. In Section 5, a brief description of the effectiveness of the proposed scheme is given in conclusion.

2. Proposed methodology

We have taken into account the 1D non-linear coupled viscous Burgers' equation. Let us consider here that the number of grid points taken is $n$in the domain $[a, b]$, and grid points are as follows, $a = x_1 < x_2 < x_3 < \ldots\ldots\ldots. < x_n = b$. These grid points are having uniform distribution, and taken step size is $h = x_{i+1}$-x_i, in the $x - direction$.

The r^{th}approximation of $u(x,t)$ and $v(x,t)$is given as follows:

$$u_x^{(r)}(x_i) = \sum_{j=1}^{n} a_{ij}^{(r)}\, u(x_j) \textbf{(2.1)}$$

$$v_x^{(r)}(x_i) = \sum_{j=1}^{n} a_{ij}^{(r)}\, v(x_j) \quad \textbf{(2.2)}$$

By using $r = 1$ in equation (2.1) and (2.2), we will get an approximation of the derivative of the first order

$$u_x^{(1)}(x_i) = \sum_{j=1}^{n} a_{ij}^{(1)}\, u(x_j) \quad \textbf{(2.3)}$$

$$v_x^{(1)}(x_i) = \sum_{j=1}^{n} a_{ij}^{(1)}\, v(x_j) \quad \textbf{(2.4)}$$

By using $r = 2$ in equation (2.3) and (2.4), we will get the approximation of derivative of second-order

$$u_x^{(2)}(x_i) = \sum_{j=1}^{n} a_{ij}^{(2)}\, u(x_j) \quad \textbf{(2.5)}$$

$$v_x^{(2)}(x_i) = \sum_{j=1}^{n} a_{ij}^{(2)}\, v(x_j) \quad \textbf{(2.6)}$$

Where, $u_x^{(1)}(x_i), v_x^{(1)}(x_i)$ are the first-order partial derivative of u and v at x_i respectively and $u_x^{(2)}(x_i)$ and $v_x^{(2)}(x_i)$ are the second-order partial derivative of u and v at grid point x_i. $u(x_j)$ *and* $v(x_j)$ are functional values of u and v at specified grid points.

2.1. Cubic Hyperbolic B-spline

Let us consider that HB_i^k is the "Hyperbolic B-spline" having order k with the given node points x_i which are in uniform distribution at $a = x_1 < x_2 < x_3 < \ldots\ldots\ldots < x_n = b$. So the "cubic Hyperbolic B-spline" will form a basis for all functions of the domain $[a, b]$. The cubic Hyperbolic B-spline of order four can be defined as the follows:

$$HB_i^4(x) = \begin{cases} \left[\dfrac{[\sinh(x-x_i)]^3}{\sinh(x_{i+3}-x_i)\sinh(x_{i+2}-x_i)\sinh(x_{i+1}-x_i)}\right], & \boldsymbol{x \in [x_i, x_{i+1}]} \\[2ex] \left[\begin{array}{l} \dfrac{[\sinh(x-x_i)]^2[\sinh(x_{i+2}-x)]}{\sinh(x_{i+3}-x_i)\sinh(x_{i+2}-x_i)\sinh(x_{i+1}-x_i)} + \\ \dfrac{\sinh(x-x_i)\sinh(x_{i+3}-x)\sinh(x-x_{i+1})}{\sinh(x_{i+3}-x_i)\sinh(x_{i+3}-x_{i+1})\sinh(x_{i+2}-x_{i+1})} + \\ \dfrac{\sinh(x_{i+4}-x)[\sinh(x-x_{i+1})]^2}{\sinh(x_{i+4}-x_{i+1})\sinh(x_{i+3}-x_{i+1})\sinh(x_{i+2}-x_{i+1})} \end{array}\right], & \boldsymbol{x \in [x_{i+1}, x_{i+2}]} \\[2ex] \left[\begin{array}{l} \dfrac{\sinh(x-x_i)[\sinh(x_{i+3}-x)]^2}{\sinh(x_{i+3}-x_i)\sinh(x_{i+3}-x_{i+1})\sinh(x_{i+3}-x_{i+2})} + \\ \dfrac{\sinh(x_{i+4}-x)\sinh(x-x_{i+1})\sinh(x_{i+3}-x)}{\sinh(x_{i+4}-x_{i+1})\sinh(x_{i+3}-x_{i+1})\sinh(x_{i+3}-x_{i+2})} + \\ \dfrac{[\sinh(x_{i+4}-x)]^2\sinh(x-x_{i+2})}{\sinh(x_{i+4}-x_{i+1})\sinh(x_{i+4}-x_{i+2})\sinh(x_{i+3}-x_{i+2})} \end{array}\right], & \boldsymbol{x \in [x_{i+2}, x_{i+3}]} \\[2ex] \left[\dfrac{[\sinh(x_{i+4}-x)]^3}{\sinh(x_{i+4}-x_{i+1})\sinh(x_{i+4}-x_{i+2})\sinh(x_{i+4}-x_{i+3})}\right], & \boldsymbol{x \in [x_{i+3}, x_{i+4}]} \\ 0, & \textbf{\textit{otherwise}} \end{cases}$$

(2.7)

Values ofHyperbolic B-spline of order four i.e. $HB_{i,4}$ ***(x) and*** $HB_{i,4}{}'$***(x) at different node points***

	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}
$HB_{i,4}(x)$	*0*	*A*	*B*	*C*	*0*
$HB'_{i,4}(x)$	*0*	*D*	*0*	*F*	*0*

Where,

$$A = \frac{[\sinh(h)]^2}{\sinh(2h)\sinh(3h)},\ B = \frac{2\sinh(h)}{\sinh(3h)},\ C = \frac{[\sinh(h)]^2}{\sinh(2h)\sinh(3h)}$$
$$D = \frac{3}{2\sinh(3h)},\ E = 0,\ F = \frac{-3}{2\sinh(3h)}$$

Where, $\{HB_0(x), HB_1(x), \dots\dots\dots\dots, HB_N(x), HB_{N+1}(x)\}$ forms a basis over given domain. In order to improve the results, modified cubic Hyperbolic B-spline can be implemented by using equation (2.7) in the way so that the obtained matrix system will become diagonally dominant [33]. Where, by using following set of equations improvised values can be obtained.

$$\begin{gathered}\emptyset_1(x) = HB_1(x) + 2\,HB_0(x) \\ \emptyset_2(x) = HB_2(x) - HB_0(x) \\ \emptyset_j(x) = HB_j(x),\ \ (j = 3,4,5,\dots\dots\dots,N-2) \\ \emptyset_{N-1}(x) = HB_{N-1}(x) - HB_{N+1}(x) \\ \emptyset_N(x) = HB_N(x) + 2\,HB_{N+1}(x)\end{gathered} \quad \textbf{(2.8)}$$

2.2. Determination of weighting coefficients

From equation (2.3) and (2.4), we have the formula to approximate first-order derivative i.e. given as follows,

$$\emptyset_k^{(1)}(x_i) = \sum_{j=1}^{n} a_{ij}^{(1)} \emptyset_k(x_j) \quad \textbf{(2.9)}$$

At grid point x_1***:*** By applying the formulas (2.7) and (2.8) in the equation (2.9) we will get the following set of equations, for different values of k,

For k=1: $\emptyset'_1(x_1) = \sum_{j=1}^{n} a_{1j}^{(1)} \emptyset_1(x_j) = a_{11}^{(1)}[B + 2C] + a_{12}^{(1)}[C]$

For k=2: $\emptyset'_2(x_1) = \sum_{j=1}^{n} a_{1j}^{(1)} \emptyset_2(x_j) = a_{11}^{(1)}[A - C] + a_{12}^{(1)}[B] + a_{13}^{(1)}[C]$

For k=3: $\emptyset'_3(x_1) = \sum_{j=1}^{n} a_{1j}^{(1)} \emptyset_3(x_j) = a_{12}^{(1)}[A] + a_{13}^{(1)}[B] + a_{14}^{(1)}[C]$

For k=4: $\emptyset'_4(x_1) = \sum_{j=1}^{n} a_{1j}^{(1)} \emptyset_4(x_j) = a_{13}^{(1)}[A] + a_{14}^{(1)}[B] + a_{15}^{(1)}[C]$

…………………………………………………
…………………………………………………
…………………………………………………

For k=n-2: $\emptyset'_{n-2}(x_1) = \sum_{j=1}^{n} a_{1j}^{(1)} \emptyset_{n-2}(x_j) = a_{1,n-3}^{(1)}[A] + a_{1,n-2}^{(1)}[B] + a_{1,n-1}^{(1)}[C]$

For k=n-1: $\emptyset'_{n-1}(x_1) = \sum_{j=1}^{n} a_{1j}^{(1)} \emptyset_{n-1}(x_j) = a_{1,n-2}^{(1)}[A] + a_{1,n-1}^{(1)}[B] + a_{1,n}^{(1)}[C-A]$

For k=n: $\emptyset'_{n}(x_1) = \sum_{j=1}^{n} a_{1j}^{(1)} \emptyset_{n}(x_j) = a_{1,n-1}^{(1)}[A] + a_{1,n}^{(1)}[B+2A]$

From the above set of the equation at grid point x_1 and for the values of $k = 1, 2, 3, \dots \dots, n$, we will obtain the following tridiagonal system of algebraic equations:

$$A \quad \vec{a}^{(1)}[i] = \vec{V}[i], \text{ where } i = 1, 2, ,3, \dots \dots, n$$

$$A = \begin{pmatrix} B+2C & C & & & & & & \\ A-C & B & C & & \cdots & & & \\ & A & B & C & & & & \\ & \vdots & & & \ddots & & \vdots & \\ & & & & & A & B & C & 0 \\ & & & & \cdots & & A & B & C-A \\ & & & & & & & A & B+2A \end{pmatrix}$$

$$\vec{a}^{(1)}[1] = \begin{pmatrix} a_{1,1}^{(1)} \\ a_{1,2}^{(1)} \\ a_{1,3}^{(1)} \\ \vdots \\ \vdots \\ a_{1,N-1}^{(1)} \\ a_{1,N}^{(1)} \end{pmatrix} \text{and} \vec{V}[1] = \begin{pmatrix} \emptyset'_1(x_1) \\ \emptyset'_2(x_1) \\ \emptyset'_3(x_1) \\ \vdots \\ \vdots \\ \emptyset'_{n-1}(x_1) \\ \emptyset'_n(x_1) \end{pmatrix} = \begin{pmatrix} 2F \\ D-F \\ 0 \\ \vdots \\ \vdots \\ . \\ 0 \end{pmatrix}$$

At grid point x_2: By applying the formulas (2.7) and (2.8) in the equation (2.9), we will get the following set of equations, for different values of k, we will get the following tridiagonal system of equations,

$$\vec{a}^{(1)}[2] = \begin{pmatrix} a_{21}^{(1)} \\ a_{22}^{(1)} \\ a_{23}^{(1)} \\ \vdots \\ \vdots \\ a_{2N-1}^{(1)} \\ a_{2N}^{(1)} \end{pmatrix} \text{and} \vec{V}[2] = \begin{pmatrix} \emptyset'_1(x_2) \\ \emptyset'_2(x_2) \\ \emptyset'_3(x_2) \\ \vdots \\ \vdots \\ \emptyset'_{n-1}(x_2) \\ \emptyset'_n(x_2) \end{pmatrix} = \begin{pmatrix} F \\ 0 \\ D \\ 0 \\ 0 \\ \vdots \\ . \\ 0 \end{pmatrix}$$

At grid point x_3: By applying the formulas (2.7) and (2.8) in the equation (2.9), we will get the following set of equations, for different values of k, we will get the following tridiagonal system of equations,

$$\vec{a}^{(1)}[3] = \begin{pmatrix} a_{31}^{(1)} \\ a_{32}^{(1)} \\ a_{33}^{(1)} \\ \vdots \\ \vdots \\ a_{3N-1}^{(1)} \\ a_{3N}^{(1)} \end{pmatrix} \text{and} \vec{V}[3] = \begin{pmatrix} \emptyset_1'(x_3) \\ \emptyset_2'(x_3) \\ \emptyset_3'(x_3) \\ \vdots \\ \vdots \\ \emptyset_{n-1}'(x_3) \\ \emptyset_n'(x_3) \end{pmatrix} = \begin{pmatrix} 0 \\ F \\ 0 \\ D \\ 0 \\ \vdots \\ . \\ 0 \end{pmatrix}$$

……………………………………………………
……………………………………………………
……………………………………………………

For grid point x_{n-1}: By applying the formulas (2.7) and (2.8) in the equation (2.9), we will get the following set of equations, for different values of k, we will get following the tridiagonal structure of equations,

$$\vec{a}^{(1)}[n-1] = \begin{pmatrix} a_{n-1,1}^{(1)} \\ a_{n-1,2}^{(1)} \\ a_{n-1,3}^{(1)} \\ \vdots \\ \vdots \\ a_{n-1,N-1}^{(1)} \\ a_{n-1,N}^{(1)} \end{pmatrix} \text{and} \vec{V}[n-1] = \begin{pmatrix} \emptyset_1'(x_{n-1}) \\ \emptyset_2'(x_{n-1}) \\ \emptyset_3'(x_{n-1}) \\ \vdots \\ \vdots \\ \emptyset_{n-1}'(x_{n-1}) \\ \emptyset_n'(x_{n-1}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ . \\ \vdots \\ . \\ F \\ 0 \\ D \end{pmatrix}$$

At grid point x_n: By applying the formulas (2.7) and (2.8) in the equation (2.9), we will get the following set of equations, for different values of k, we will get the following tridiagonal system of equations,

$$\vec{a}^{(1)}[n] = \begin{pmatrix} a_{n,1}^{(1)} \\ a_{n,2}^{(1)} \\ a_{n,3}^{(1)} \\ \vdots \\ \vdots \\ a_{n,n-1}^{(1)} \\ a_{n,n}^{(1)} \end{pmatrix} \text{and} \vec{V}[n] = \begin{pmatrix} \emptyset_1'(x_n) \\ \emptyset_2'(x_n) \\ \emptyset_3'(x_n) \\ \vdots \\ \vdots \\ \emptyset_{n-1}'(x_n) \\ \emptyset_n'(x_n) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ . \\ \vdots \\ . \\ 0 \\ F-D \\ 2D \end{pmatrix}$$

Similarly, in order to find the weighting coefficients of order n where $n \geq 2$, is given as following from [37], given as follows,

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

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

So, with the help of the above-mentioned equations 2^{nd} and higher-order weighting coefficients can be easily obtained.

2.3. Implementation of method modified cubic Hyperbolic B-spline DQM

Since considered 1-D coupled non-linear Burgers' equation is given as follows:

$$u_t + \delta u_{xx} + \eta u u_x + \alpha (uv)_x = 0$$
$$v_t + \mu v_{xx} + \xi v v_x + \beta (uv)_x = 0$$

So firstly, the spatial derivatives of first and second-order given in equations (1.1) and (1.2) will be discretized by implementing the proposed scheme i.e., modified cubic Hyperbolic B-spline DQM. After this implementation, a system of ordinary differential equations (non-linear) will be obtained as follows:

$$\frac{du}{dt} = -\delta \sum_{j=1}^{n} a_{ij}^{(2)} u(x_j) - \eta u(x_i) \sum_{j=1}^{n} a_{ij}^{(1)} u(x_j) - \alpha [u(x_i) \sum_{j=1}^{n} a_{ij}^{(1)} v(x_j) + v(x_i) \sum_{j=1}^{n} a_{ij}^{(1)} u(x_j)] \quad \textbf{(2.10)}$$

$$\frac{dv}{dt} = -\mu \sum_{j=1}^{n} a_{ij}^{(2)} v(x_j) - \xi v(x_i) \sum_{j=1}^{n} a_{ij}^{(1)} v(x_j) - \beta [u(x_i) \sum_{j=1}^{n} a_{ij}^{(1)} v(x_j) + v(x_i) \sum_{j=1}^{n} a_{ij}^{(1)} u(x_j)] \quad \textbf{(2.11)}$$

This resulting system of ODEs will be solved by the SSP-RK43 scheme [55],and two types of errors have been discussed in the paper, the errors between exact solutions and the approximate solutions, given by norms,

$$L_2 = (h \sum_{j=1}^{N} |U_j^{exac} - U_j^N|^2)^{\frac{1}{2}}$$
$$L_\infty = \max_j |U_j^{exac} - U_j^N|$$

3.Numerical Experiments and Discussion

In this section, three examples are discussed for checking the effectiveness and accuracy of the proposed scheme. In Example 1, a graphical representation of the numerical solution of both u and v components is given with mentioned parameters at different time levels. In Example 1, no exact solution is given;the only discussion about the numerical solution is presented. In example 2, $L_2$and L_∞ errors are calculated for different grid points in order to check the accuracy of the present method, comparison with the previous scheme is also presented as well as this example is discussed by means of tables and graphs. In Example 3, a comparison with previous schemes is also presented given in the form of tables and figures.

Example1. Considered coupled 1-D non-linear Burgers' equation given as follows [9,10].

$$u_t + \delta u_{xx} + \eta u u_x + \alpha (uv)_x = 0$$
$$v_t + \mu v_{xx} + \xi v v_x + \beta (uv)_x = 0$$

By setting δ = -1 and μ = -1 in above equations, the following set of equations will be obtained

$$u_t - u_{xx} + \eta u u_x + \alpha (uv)_x = 0 \quad \textbf{(3.1)}$$
$$v_t - v_{xx} + \xi v v_x + \beta (uv)_x = 0 \quad \textbf{(3.2)}$$

Where,η, ξ, α, β all can be selected arbitrarily, initial conditions are given as following from [14],

$$u(x,0) = \begin{cases} \sin(2\pi x), & x \epsilon [0, 0.5] \\ 0, & x \epsilon (0.5, 1] \end{cases} \quad \textbf{(3.3)}$$

$$v(x,0) = \begin{cases} 0, & x \epsilon [0, 0.5] \\ -\sin(2\pi x), & x \epsilon (0.5, 1] \end{cases} \quad \textbf{(3.4)}$$

Boundary conditions are all zero.

In Figure 1, Numerical solutions of u and v components are shown at different t = 0.1, 0.2, 0.3, 0.4 for N = 20, $\eta = 1, \xi = 5, \alpha = 10, \beta = 10$ at Δt = 0.001.In Figure 2, graphical representation of numerical solutions of both u and v components is given at t = 0.1, 0.2, 0.3 and 0.4 for the parameters N = 20, $\eta = 10, \xi = 10, \alpha$ = 100, $\beta = 100$ and Δt = 0.001.In Figure 3, graphical representation of numerical solutions of both u and v components is given at t = 0.1, 0.2, 0.3 and 0.4 for the parameters N = 20, $\eta = 100, \xi = 100, \alpha$ = 100, $\beta = 100$ and Δt = 0.001.In Figure 4, graphical representation of numerical solutions of u component in different forms like, mesh and surface is given for the mentioned parameters in figure.In Figure 5, graphical representation of numerical solutions of v component in different forms like, mesh and surface is given for the mentioned parameters in figure.In Figure 6, graphical representation of numerical solutions of both u and v components is given at t = 0.1, 0.2, 0.3, 0.4 and 0.5 respectively, for the parameters N = 100, $\eta = 100, \xi = 100, \alpha$ = 100, $\beta = 100$ and Δt = 0.001.In Figure 7, graphical representation of numerical solutions of both u and v components is given at t = 0.1, 0.2, 0.3, 0.4 and 0.5 respectively, for the parameters N = 100, $\eta = 10, \xi = 10, \alpha$ = 100, $\beta = 100$ and Δt = 0.001.

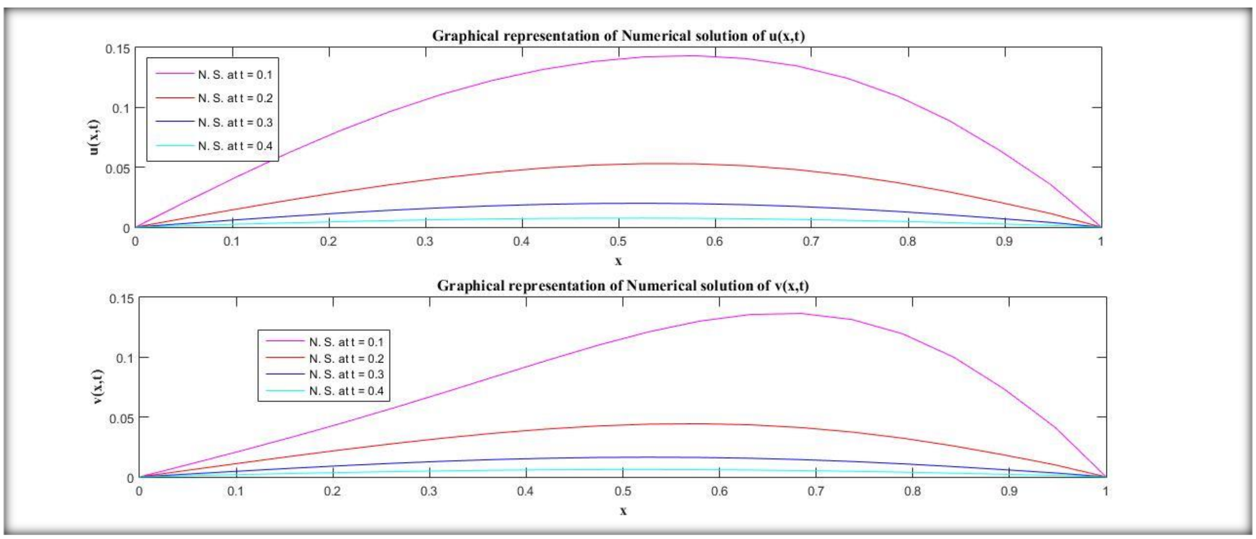

FIGURE 1.
Graphical representation of Numerical solution of u(x, t) and v(x, t) for N = 20, $\eta = 1, \xi = 5, \alpha = 10, \beta = 10$, Δt = 0.001 at time levels t = 0.1, 0.2, 0.3 and 0.4 respectively

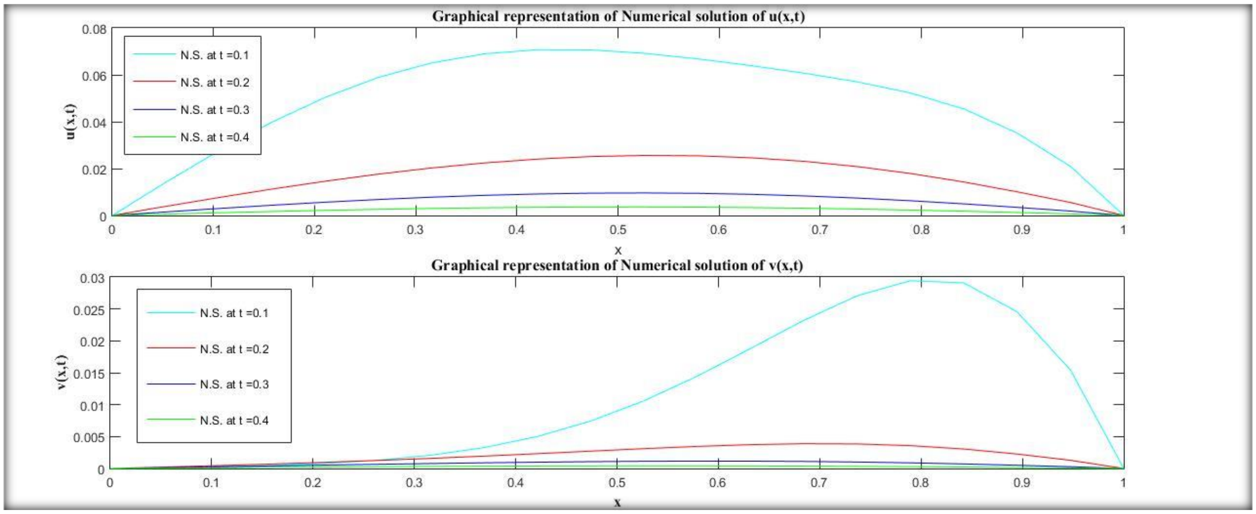

FIGURE 2.
Graphical representation of Numerical solution of u(x, t) and v(x, t) for N = 20, $\eta = 10, \xi = 10, \alpha = 100, \beta = 100$, Δt = 0.001 at time levels t = 0.1, 0.2, 0.3 and 0.4respectively

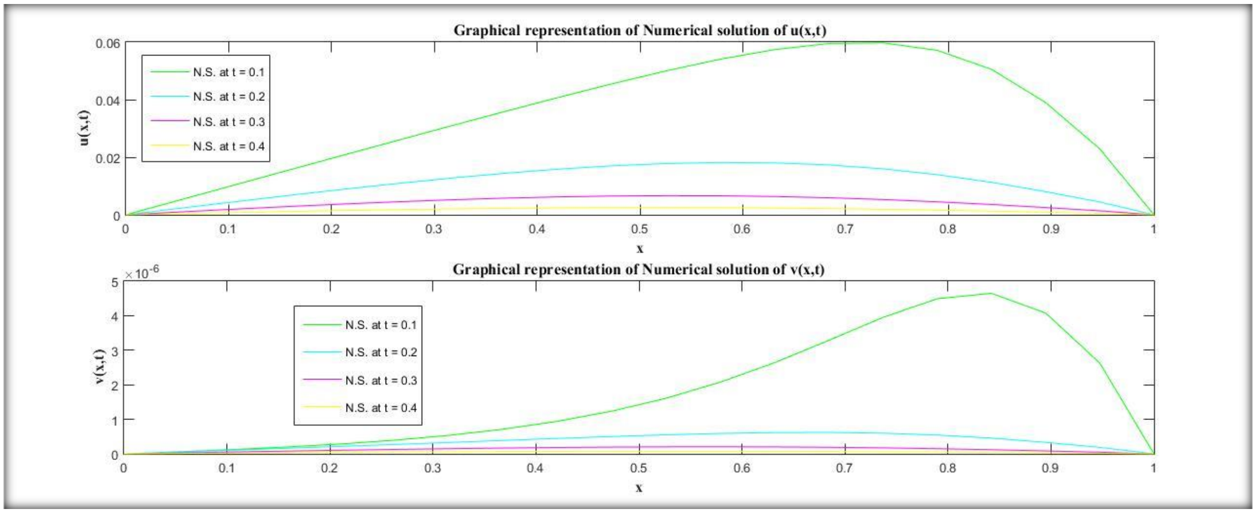

FIGURE 3.
Graphical representation of Numerical solution of u(x, t) and v(x, t) for N = 20,$\eta = 100$, $\xi = 100, \alpha = 100, \beta = 100$, Δt = 0.001 at time levels t = 0.1, 0.2, 0.3and 0.4 respectively

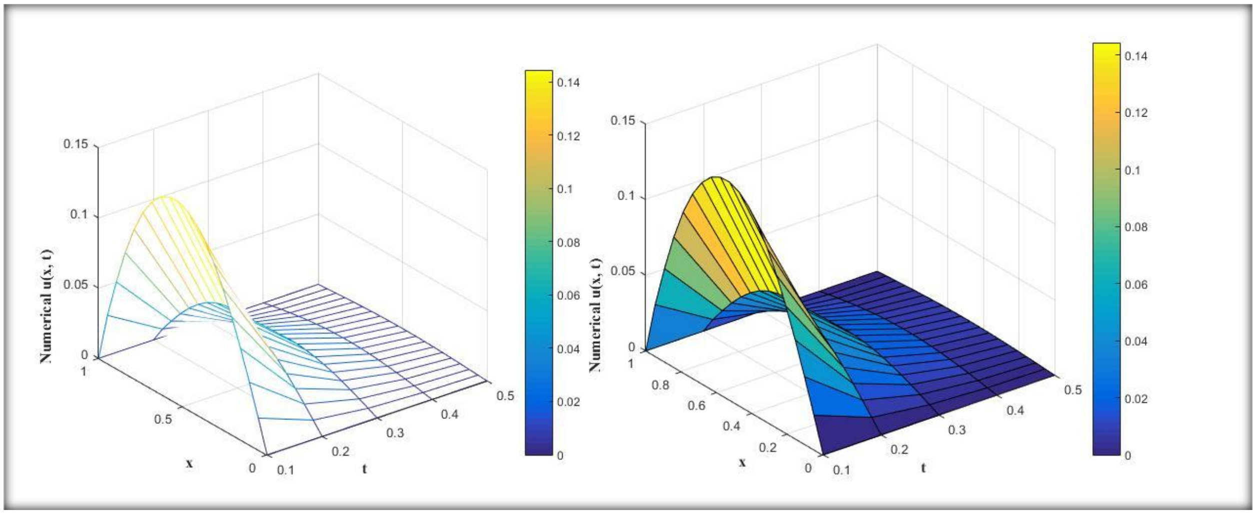

FIGURE 4.

Graphical representation of Numerical solution of u(x, t) for $\boldsymbol{\eta = 10, \xi = 10, \alpha = 10, \beta = 10}$, Δt = 0.001 at time levels t = 0.1, 0.2, 0.3, 0.4 and 0.5 respectively

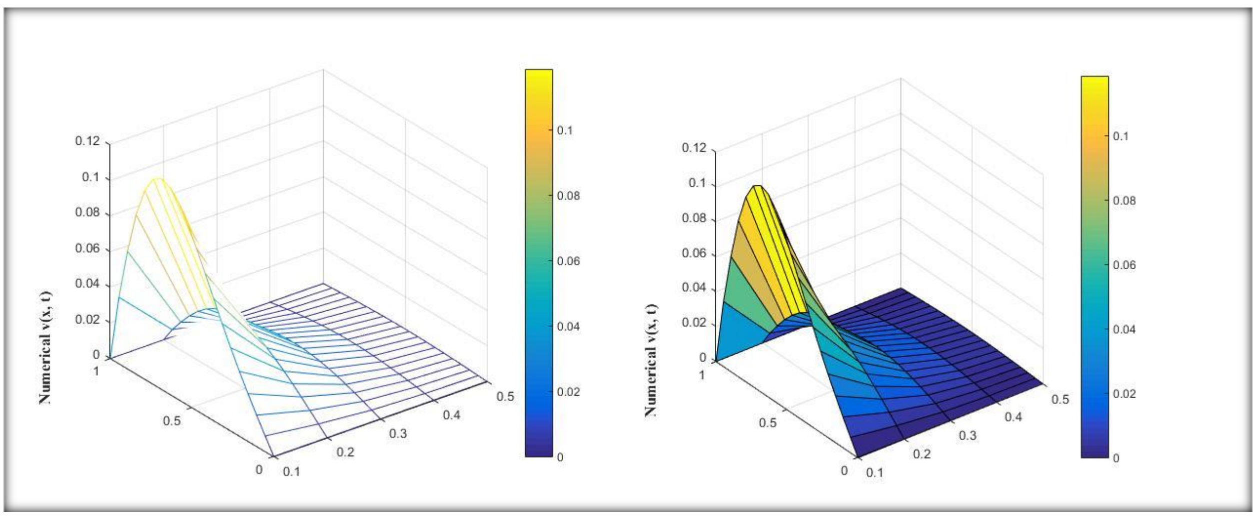

FIGURE 5.

Graphical representation of Numerical solution of v(x, t) for$\boldsymbol{\eta = 10, \xi = 10, \alpha = 10, \beta = 10}$, Δt = 0.001 at time levels t = 0.1, 0.2, 0.3, 0.4and 0.5 respectively

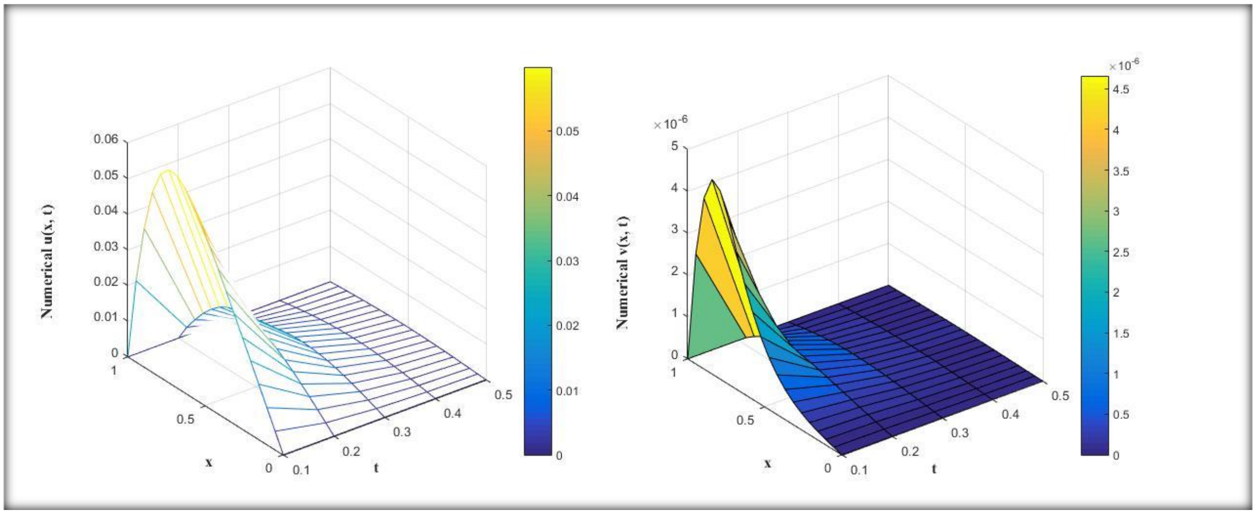

FIGURE 6.
Graphical representation of Numerical solution of u(x, t) and v(x, t) for $\boldsymbol{\eta = 100, \xi = 100, \alpha = 100, \beta = 100}$, ***Δt = 0.001 at time levels t = 0.1, 0.2, 0.3, 0.4 and 0.5 respectively***

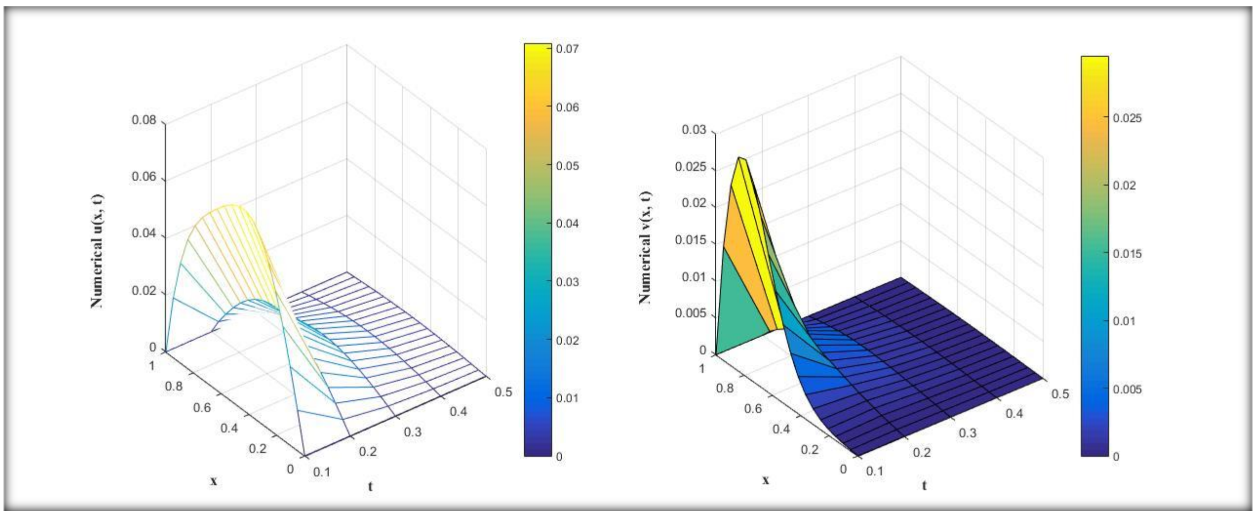

FIGURE 7.
Graphical representation of Numerical solution of u(x, t) and v(x, t) for $\boldsymbol{\eta = 10, \xi = 10, \alpha = 100, \beta = 100}$, ***Δt = 0.001 at time levels t = 0.1, 0.2, 0.3,0.4 and 0.5 respectively***

Example 2. By setting the parameters $\delta = -1, \mu = -1$ is equations (1.1) and (1.2), following coupled equation will be formed given as follows,

$$u_t = u_{xx} - \eta u u_x - \alpha(uv)_x \quad \textbf{(3.5)}$$
$$v_t = v_{xx} - \xi v v_x - \beta(uv)_x \quad \textbf{(3.6)}$$

Exact solution was given by Kaya[2] for equations (3.5) and (3.6) is given as follows,

$$u(x,t) = a_0 - 2A\,\frac{(2\,\alpha-1)}{(4\,\alpha\,\beta-1)}\tanh\,[A(x-2At)] \quad \textbf{(3.7)}$$
$$v(x,t) = a_0\,\frac{2\,\beta-1}{2\,\alpha-1} - 2A\,\frac{(2\,\alpha-1)}{(4\,\alpha\,\beta-1)}\tanh[A(x-2At)] \quad \textbf{(3.8)}$$

Where, $A = a_0 \frac{(4\,\alpha\,\beta-1)}{(2\,\alpha-1)}$,
Domain of computation is given as $x \in [-10, 10], t > 0$

Initial Condition:

$$u(x,0) = a_0 - 2A\frac{(2\,\alpha-1)}{(4\,\alpha\,\beta-1)}\tanh[Ax] \quad \textbf{(3.9)}$$

$$v(x,0) = a_0\frac{2\,\beta-1}{2\,\alpha-1} - 2A\frac{(2\,\alpha-1)}{(4\,\alpha\,\beta-1)}\tanh[Ax] \quad \textbf{(3.10)}$$

Boundary Conditions:

$$u(-10,t) = a_0 - 2A\frac{(2\,\alpha-1)}{(4\,\alpha\,\beta-1)}\tanh[A(-10-2At)] \quad \textbf{(3.11)}$$

$$u(10,t) = a_0 - 2A\frac{(2\,\alpha-1)}{(4\,\alpha\,\beta-1)}\tanh[A(10-2At)] \quad \textbf{(3.12)}$$

$$v(-10,t) = a_0\frac{(2\,\beta-1)}{(2\,\alpha-1)} - 2A\frac{(2\,\alpha-1)}{(4\,\alpha\,\beta-1)}\tanh[A(-10-2At)] \quad \textbf{(3.13)}$$

$$v(10,t) = a_0\frac{(2\,\beta-1)}{(2\,\alpha-1)} - 2A\frac{(2\,\alpha-1)}{(4\,\alpha\,\beta-1)}\tanh[A(10-2At)] \quad \textbf{(3.14)}$$

In following Table 1, L_2 and L_∞ errors are presented for various grid points.In Table 1, it has been observed that on an increasing number of grid points and with the mentioned parameters in this Table 1, L_2 *and* L_∞ error norms for both u and v components got reduced**.**In following Table 2, L_2 and L_∞ errors are presented for a different number of grid points. In Table 2, for the mentioned parameters in the table, on the increasing number of grid points, L_2 and L_∞ error norms got reduced for both u and v components. In Table 3, a comparison of errors is given. In Table 3, a comparison with Raslan et al. [17] is given, obtained results in the present scheme for the mentioned parameters in the table are in good agreement with the compared results. In Table 4, a comparison of errors for the u-component has been made with Raslan et al. [17], Khater et al. [6], and Rashid and Ismail [8]. In Table 4, results obtained by the present method are in good compatibility on making a comparison with previous ones. In Table 5, a comparison of errors for v-component has been made with Raslan et al. [17], Khater et al. [6], and Rashid and Ismail [8]. In Table 5, on making this comparison with previous results, it can be observed that the proposed scheme is producing acceptable results. In Table 6, a comparison of errors for the u-component has been made with Raslan et al. [17], Mittal and Jiwari [11], and Mittal and Arora [14]. In Table 7, a comparison of errors for v-component has been made with Raslan et al. [17], Mittal and Jiwari [11], and Mittal and Arora [14]. In Figures 8, 9, 10,11, 12, and 13, Graphical representations of exact and numerical solutions are given for u, and v components are given for mentioned parameters. In Figure 8,comparison between exact and numerical solutions of u and v components is given for $\Delta t = 0.001, \alpha = 0.1$, $\beta = 0.3$, $N = 10$, $a_0 = 0.05$ at time level $t = 0.5$. In Figure 9 comparison between exact and numerical solutions of u and v components is given for $\Delta t = 0.001, \alpha = 0.3$, $\beta = 0.3$, $a_0 = 0.05$ $N = 10$, $a_0 = 0.05$ at time level $t = 0.5$. In Figure 10 comparison between exact and numerical solutions of u and v components is given for $\Delta t = 0.001, \alpha = 0.1$, $\beta = 0.3$, $N = 20$, $a_0 = 0.05$ at time level $t = 1$. In Figure 11 comparison between exact and numerical solutions of u and v components is given for $\Delta t = 0.001, \alpha = 0.3$, $\beta = 0.3$, $N = 20$, $a_0 = 0.05$ at time level $t = 1$. In Figure 12 both u and v components are compared for exact and numerical solutions for the parameters $\Delta t = 0.001, \alpha = 0.1$, $\beta = 0.3$, $a_0 = 0.05$, $N =$

20 at the time levels t = 0.5, 1, 1.5, 2, 2.5. In Figure 13 both u and v components are compared for exact and numerical solutions for the parameters $\Delta t = 0.001, \alpha = 0.1,\ \beta = 0.5, a_0 = 0.05, N = 30$ at the time levels t = 0.5, 1, 1.5, 2, 2.5. From all of these figures, it is quite obvious that the results produced by the present scheme are in a good match with an exact solution.

TABLE 1.
L_2 and L_∞ errors for u component and v component for $\Delta t = 0.001$, $a_0 = 0.05$, $\alpha = 0.1$, $\beta = 0.5$ at time level t = 0.5 for different grid points

	u-component		*v-component*	
Number of grid points	L_2 Error Norm	L_∞ Error Norm	L_2 Error Norm	L_∞ Error Norm
20	3.5244E-4	1.7189E-4	5.9022E-5	3.4779E-5
30	3.5972E-4	1.7127E-4	5.9590E-5	4.1158E-5
50	3.4735E-4	1.3212E-4	5.5250E-5	3.3702E-5
100	3.3457E-4	8.5068E-5	5.3022E-5	2.0973E-5
120	3.3258E-4	8.5077E-5	5.3037E-5	2.1048E-5
150	3.3069E-4	8.5084E-5	5.3206E-5	2.1121E-5
180	3.2950E-4	8.5086E-5	5.3411E-5	2.1178E-5
200	3.2892E-4	8.5087E-5	5.3542E-5	2.1205E-5
220	3.2847E-4	8.5087E-5	5.3664E-5	2.1229E-5
250	3.2793E-4	8.5087E-5	5.3829E-5	2.1260E-5

TABLE 2.
L_2 and L_∞ errors for u component and v component for $\Delta t = 0.001$, $a_0 = 0.005$, $\alpha = 0.1$, $\beta = 0.5$ at time level t = 0.5 for different grid points

	u-component		*v-component*	
Number of grid points	L_2 Error Norm	L_∞ Error Norm	L_2 Error Norm	L_∞ Error Norm
20	2.0508E-6	1.7716E-6	6.3472E-7	6.0000E-7
30	1.9368E-6	1.8783E-6	5.8195E-7	6.3358E-7
50	1.5344E-6	1.4926E-6	4.2106E-7	5.0148E-7
100	1.1437E-6	8.8060E-7	2.4396E-7	2.9510E-7
120	1.0828E-6	7.5180E-7	2.1082E-7	2.5182E-7
150	1.0266E-6	6.7947E-7	1.7692E-7	2.061E-7
180	9.9248E-7	6.8067E-7	1.5400E-7	1.7430E-7
200	9.7668E-7	6.8108E-7	1.4249E-7	1.5800E-7
220	9.6445E-7	6.8163E-7	1.3307E-7	1.4447E-7
250	9.5067E-7	6.8163E-7	1.2178E-7	1.2801E-7

TABLE 3.
Comparison of L_2 and L_∞ errors for $\Delta t = 0.01$, $\eta = 2$, $\xi = 2$, $\alpha = 0.1$ and $\beta = 0.3$ at time level t = 1 for different grid points

	Raslan et al. [17]	Raslan et al. [17]	Present Method	Present Method

	u-component		v-component		u-component		v-component	
N	L_2 Error Norm	L_∞ Error Norm	L_2 Error Norm	L_∞ Error Norm	L_2 Error Norm	L_∞ Error Norm	L_2 Error Norm	L_∞ Error Norm
10	2.92E-4	9.02E-5	1.11E-4	4.54E-5	3.73E-4	1.29E-4	1.53E-4	5.43E-5
50	3.01E-4	8.23E-5	1.11E-4	4.19E-5	7.06E-4	1.81E-4	3.82E-4	1.08E-4
100	3.01E-4	8.24E-5	1.12E-4	4.19E-5	6.91E-4	1.81E-4	3.73E-4	1.09E-4
200	3.02E-4	8.20E-5	1.16E-4	4.20E-5	6.83E-4	1.81E-4	3.69E-4	1.09E-4

TABLE 4.

Comparison of errors for u components with Raslan et al. [17], Khater et al. [6] and Rashid and Ismail [8] for the parameters α_0 = 0.05, N = 16, Δt = 0.01 at different time levels

			Raslan et al. [17]		Khater et al. [6]	Rashid and Ismail [8]	Present Method	
t	α	β	L_2	L_∞	L_∞	L_∞	L_2	L_∞
.5	.1	.3	1.51E-4	4.43E-5	1.44E-3	9.62E-4	3.57E-4	1.83E-4
.5	.3	.3	2.06E-4	6.42E-5	-	-	5.21E-4	2.32E-4
1	.1	.3	2.97E-4	8.44E-5	1.27E-3	1.15E-3	6.57E-4	2.59E-4
1	.3	.3	4.07E-4	1.19E-4	-	-	9.86E-4	3.27E-4

TABLE 5.

Comparison of errors for v components with Raslan et al. [17], Khater et al. [6] and Rashid and Ismail [8] for the parameters α_0= 0.05, N = 16, Δt = 0.01 at different time levels

			Raslan et al. [17]		Khater et al. [6]	Rashid and Ismail [8]	Present Method	
t	α	β	L_2	L_∞	L_∞	L_∞	L_2	L_∞
0.5	0.1	0.3	5.77E-4	2.34E-5	5.42E-4	3.33E-4	1.94E-4	1.07E-4
0.5	0.3	0.3	2.06E-4	6.42E-5	-	-	5.21E-4	2.32E-4
1	0.1	0.3	1.13E-4	4.42E-5	1.29E-3	1.16E-3	3.42E-4	1.36E-4
1	0.3	0.3	4.07E-4	1.19E-4	-	-	9.86E-4	3.27E-4

TABLE 6.

Comparison of errors for u components with Raslan et al. [17], Mittal and Jiwari [11] and Mittal and Arora [14] for the parameters α_0= 0.05, N = 21, η = 2 , ξ = 2, Δt = 0.01 at different time levels

			Raslan et al. [17]		Mittal and Jiwari[11]	Mittal and Arora [14]	Present Method	
t	α	β	L_2	L_∞	L_∞	L_∞	L_2	L_∞
0.5	0.1	0.3	1.52E-4	4.33E-5	4.17E-5	4.17E-5	3.90E-4	2.12E-4
0.5	0.3	0.3	2.07E-4	6.10E-5	-	-	2.12E-4	2.65E-4

1	0.1	0.3	2.98E-4	8.17E-5	8.28E-5	8.26E-5	7.10E-4	7.10E-4
1	0.3	0.3	4.09E-4	1.17E-4	-	-	1.08E-3	3.40E-4

TABLE 7.

Comparison of errors for v component with Raslan et al. [17], Mittal and Jiwari [11] and Mittal and Arora [14] for the parameters α_0= 0.05, N = 21, η = 2 , ξ = 2, Δt = 0.01 at different time levels

			Raslan et al. [17]		Mittal and Jiwari [11]	Mittal and Arora [14]	Present Method	
t	α	β	L_2	L_∞	L_∞	L_∞	L_2	L_∞
0.5	0.1	0.3	5.82E-5	2.32E-5	5.42E-5	1.48E-4	2.16E-4	1.26E-4
0.5	0.3	0.3	2.07E-4	6.10E-5	-	-	5.73E-4	2.65E-4
1	0.1	0.3	1.14E-4	4.15E-5	1.07E-4	4.77E-4	3.78E-4	1.45E-4
1	0.3	0.3	4.09E-4	1.17E-4	-	-	1.08E-3	3.40E-4

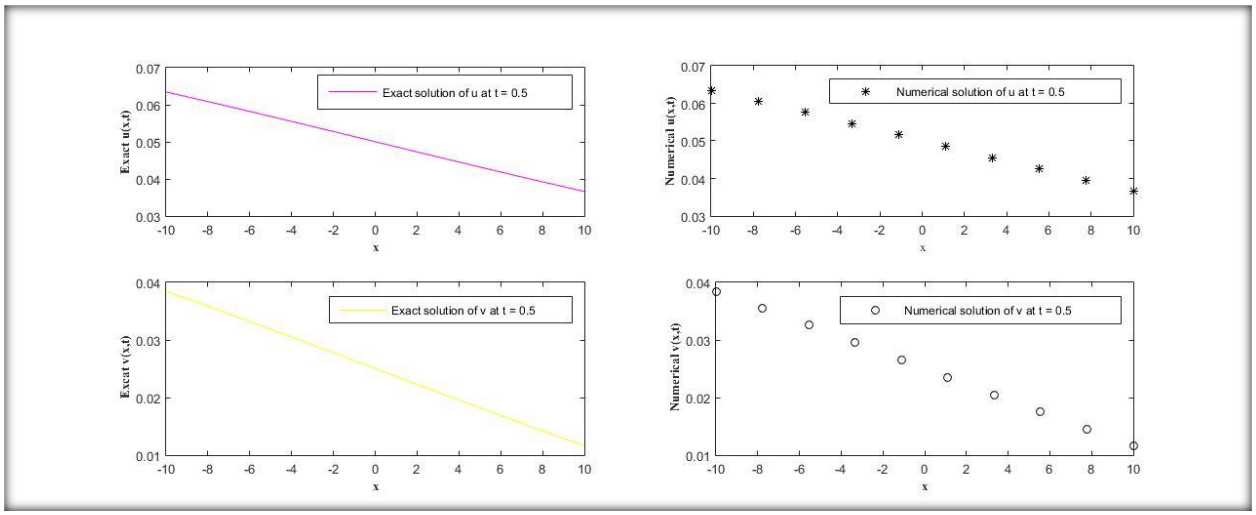

FIGURE 8.

Comparison between Exact and Numerical solution of u(x, t) and v(x, t) for parameters $\Delta t = 0.001, a_0 = 0.05$, $N = 10, \alpha = 0.1$, β = 0.3 at time level t = 0.5

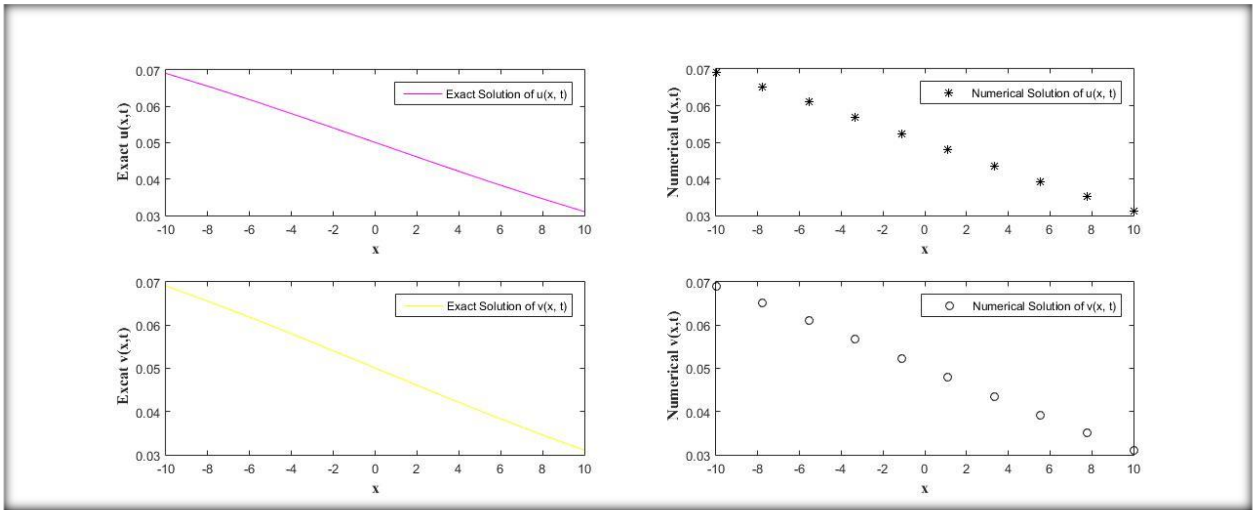

FIGURE 9.
Comparison of Exact and Numerical solutions of u and v components for parameters parameters$\Delta t = 0.001$, $N = 10, \alpha = 0.3$, $\beta = 0.3$ at time level t = 0.5

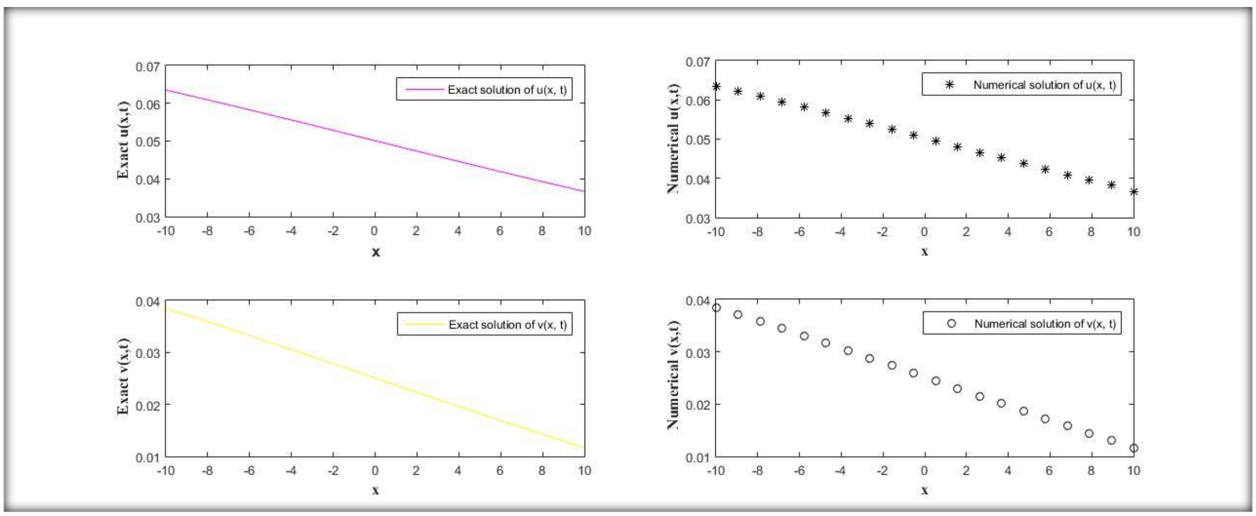

FIGURE 10.
Comparison between Exact and Numerical solutions of u and v components for parameters parameters$\Delta t = 0.001$, $N = 20, \alpha = 0.1$, $\beta = 0.3$ at time level t = 1

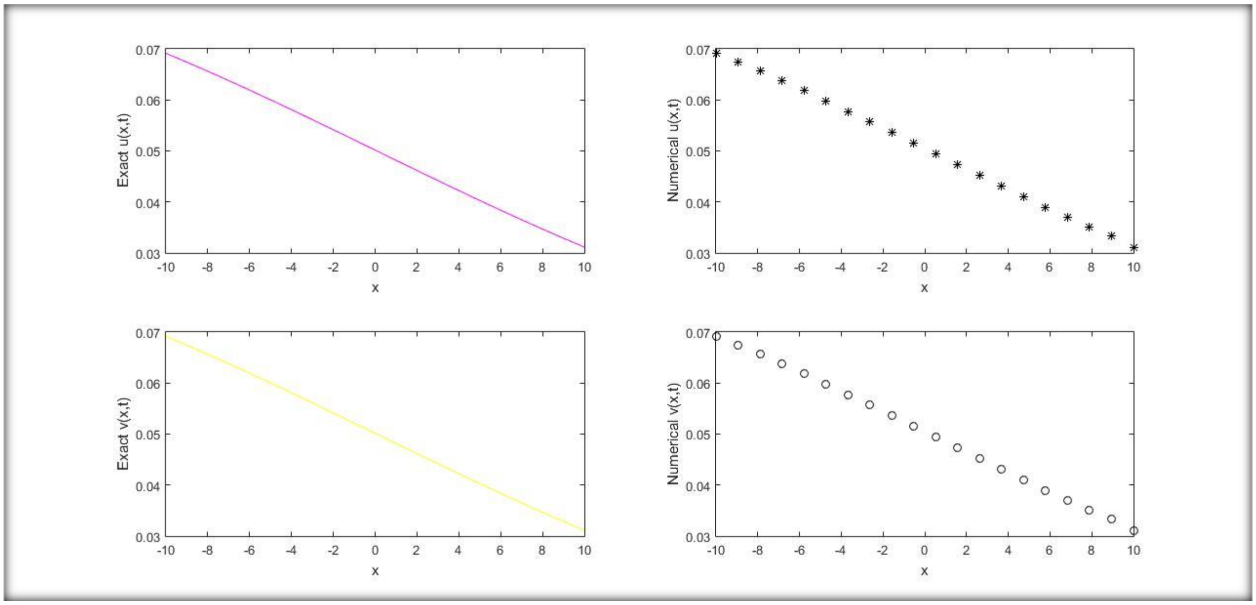

FIGURE 11.
Comparison between Exact and Numerical solutions of u and v components for parameters parameters$\Delta t = 0.001$, $N = 20, \alpha = 0.3$, $\beta = 0.3$ at time level t = 1

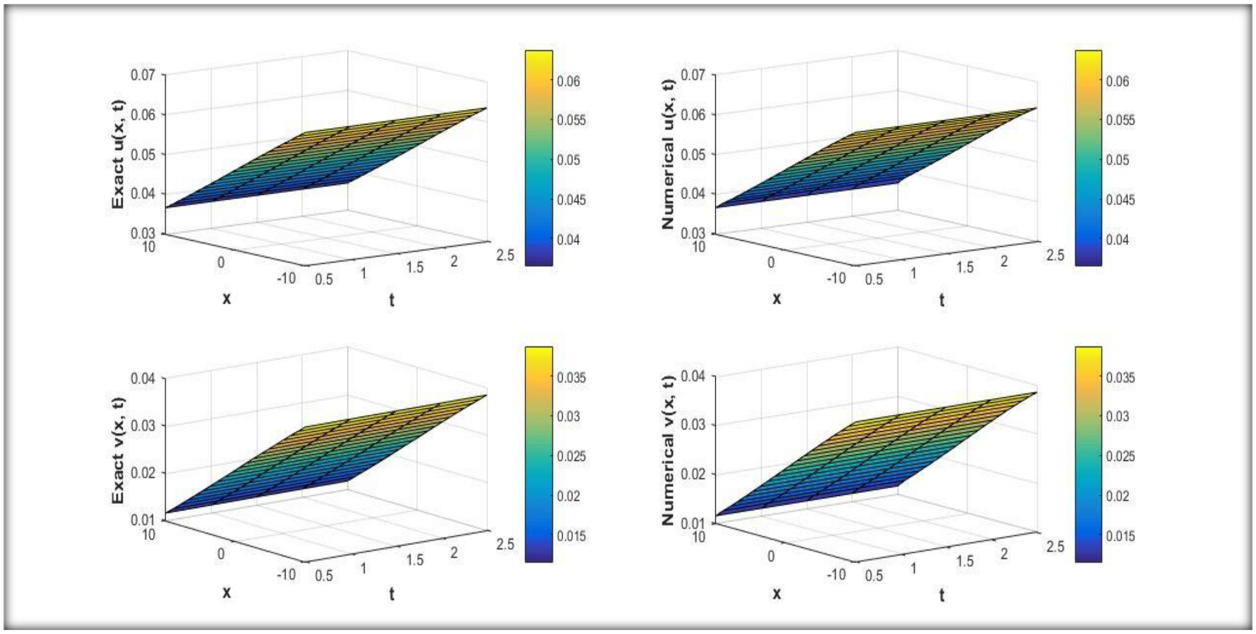

FIGURE 12.
Graphical representation of Exact and Numerical solutions of u and v components for parameters parameters $\Delta t = 0.001$, $N = 20, \alpha = 0.1$, $\beta = 0.3$ at time level t = 0.5, 1, 1.5, 2 and 2.5

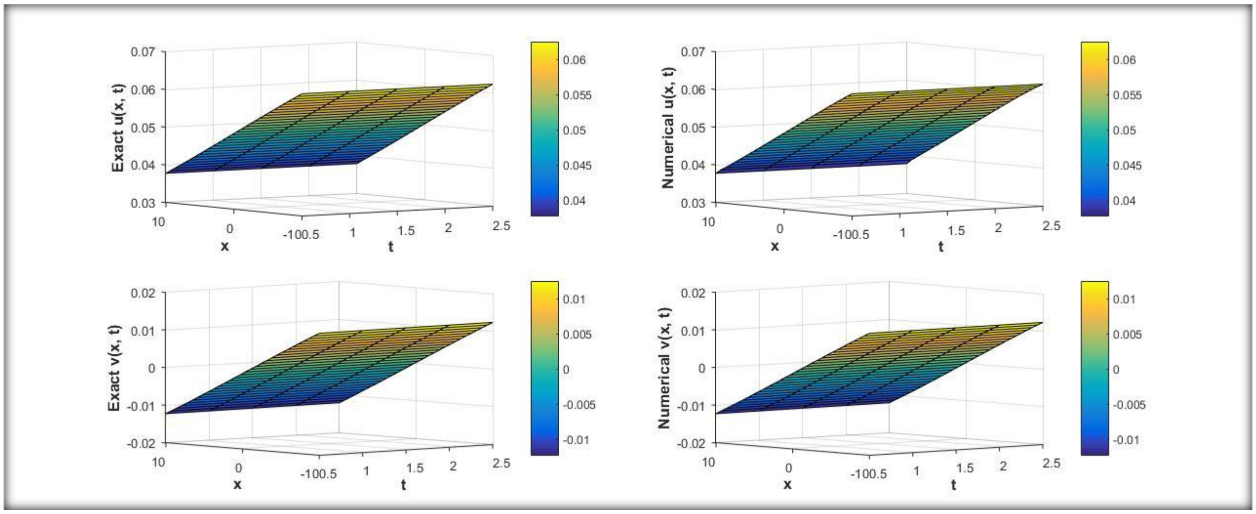

FIGURE 13.
Graphical representation of Exact and Numerical solutions of u and v components for parameters parameters$\Delta t = 0.001$, $N = 30$, $\alpha = 0.1$, $\beta = 0.5$, $a_0 = 0.5$ at time level t = 0.5, 1, 1.5, 2 and 2.5

Example 3. By using the parameters $\delta = -1, \mu = -1, \eta = -2,\ \xi = -2,\ \alpha = \frac{5}{2} and\ \beta = \frac{5}{2}$ in coupled 1-D equation (1.1) and (1.2), the following system of coupled equation will be obtained,

$$u_t - u_{xx} - 2\, uu_x + \frac{5}{2}(uv)_x = 0 \quad \textbf{(3.15)}$$

$$v_t - v_{xx} - 2\, vv_x + \frac{5}{2}(uv)_x = 0 \quad \textbf{(3.16)}$$

The analytical solutionof the above-coupled equations ispresented as follows:

$$\begin{cases} u(\mathrm{x},\mathrm{t}) = \lambda\left[1 - \tanh\left\{\frac{3}{2}\,\lambda\,(x - 3\,\lambda\,t)\right\}\right] \\ v(\mathrm{x},\mathrm{t}) = \lambda\,[1 - \tanh\{\frac{3}{2}\,\lambda\,(x - 3\,\lambda\,t)\}] \end{cases},\quad x \in [-20,20] \text{ and } t > 0 \textbf{(3.17)}$$

Initial conditions:

$$u(x,0) = \lambda\left[1 - \tanh\left\{\frac{3}{2}\,\lambda\,x\right\}\right] \textbf{(3.18)}$$

$$v(x,0) = \lambda\left[1 - \tanh\left\{\frac{3}{2}\,\lambda\,x\right\}\right] \textbf{(3.19)}$$

Boundary Conditions:

$$u(-20,t) = \lambda\left[1 - \tanh\left\{\frac{3}{2}\,\lambda\,(-20 - 3\,\lambda\,t)\right\}\right] \textbf{(3.20)}$$

$$u(20,t) = \lambda\left[1 - \tanh\left\{\frac{3}{2}\,\lambda\,(20 - 3\,\lambda\,t)\right\}\right] \textbf{(3.21)}$$

$$v(-20,t) = \lambda\,[1 - \tanh\{\frac{3}{2}\,\lambda\,(-20 - 3\,\lambda\,t)\}] \textbf{ (3.22)}$$

$$v(20,t) = \lambda\,[1 - \tanh\{\frac{3}{2}\,\lambda\,(20 - 3\,\lambda\,t)\}] \textbf{(3.33)}$$

In Table 8 and Table 9, a comparison of L_2 and L_∞Errors is given for u-component with [13] for the parameters Δt = 0.001, N=320at t = 1,2 ,3, 4 and 5 for the values of λ = 0.1 and λ =0.5 respectively. Errors are not compared for v-component because it can be understood that u and v both components are same. In Figure 14, graphical representation of exact and numerical solutions of u-component is given for Δt = 0.001, N = 101 at t = 1, 5, 10, 15, 20 and 25 for the value of λ = 0.1. In Figure 15, graphical representation of exact and numerical solutions of u component is given for Δt = 0.001, N = 320 at t = 1, 3, 5, 7, 9 and 11 for the value of λ = 0.5. In following Figure 16, graphical representation of exact and numerical solutions of u- component is given for Δt = 0.001, N = 320 at t = 1, 2, 3, 4 and 5 for the value of λ = 0.1. In Figure 17, a comparison between exact and numerical solution of $u(x,t)$ is given for the parameters N = 320, Δt = 0.001, λ = 0.5 at the t = 1, 2, 3, 4 and 5 respectively. In Figure 18, comparison of exact and numerical solutions is given for N= 350, Δt = 0.001, λ = 0.1 at Figure 19, analysis of exact and numerical solutions of $u(x,t)$ is done with the help of graphs for N = 350, Δt =0.001, λ = 0.5 at t = 0.1, 0.2, 0.3, 0.4 and 0.5.

TABLE 8.
Errors compared with LBM[13] and FDM [13] with parameters Δt = 0.001, N = 320 and λ= 0.1 for u-component

	L_2 Error Norm			L_∞ Error Norm		
t	***LBM [13]***	***FDM [13]***	***Present***	***LBM [13]***	***FDM [13]***	***Present***
1	*1.4829 × 10−6*	*1.5724 × 10−6*	*1.94E-6*	*5.7788 × 10−7*	*6.2856×10−7*	*2.58E-6*
2	*2.7955 × 10−6*	*2.9383 × 10−6*	*2.51E-6*	*1.0754 × 10−6*	*1.1138×10−6*	*2.87E-6*
3	*3.9298 × 10−6*	*4.1676 × 10−6*	*3.01E-6*	*1.4861 × 10−6*	*1.5879×10−6*	*3.17E-6*
4	*4.9434 × 10−6*	*5.2504 × 10−6*	*3.50E-6*	*1.8800 × 10−6*	*1.9868×10−6*	*3.48E-6*
5	*5.8615 × 10−6*	*6.1878 × 10−6*	*3.99E-6*	*2.2034 × 10−6*	*2.3468×10−6*	*3.82E-6*

TABLE 9.
Comparison of L_2 andL_∞errors for u-component for the parameters Δt = 0.001, N = 320, λ= 0.5 at the time levels t = 1, 2, 3, 4 and 5

	L_2 Error Norm			L_∞ Error Norm		
Time level	***LBM [13]***	***FDM [13]***	***Present***	***LBM [13]***	***FDM [13]***	***Present***
1	*1.6362 × 10−4*	*1.6930 × 10−4*	*1.28E-5*	*6.7505 × 10−4*	*7.1670×10−4*	*1.15E-5*
2	*1.9746 × 10−4*	*2.0052 × 10−4*	*2.26E-5*	*8.1705 × 10−4*	*8.6540×10−4*	*1.93E-5*
3	*2.0557 × 10−4*	*2.0625 × 10−4*	*3.19E-5*	*8.6375 × 10−4*	*9.1450×10−4*	*2.65E-5*
4	*2.0543 × 10−4*	*2.0466 × 10−4*	*4.11E-5*	*8.8160 × 10−4*	*9.3330×10−4*	*3.34E-5*
5	*2.0231 × 10−4*	*2.0074 × 10−4*	*5.03E-5*	*8.9060 × 10−4*	*9.4110×10−4*	*4.03E-5*

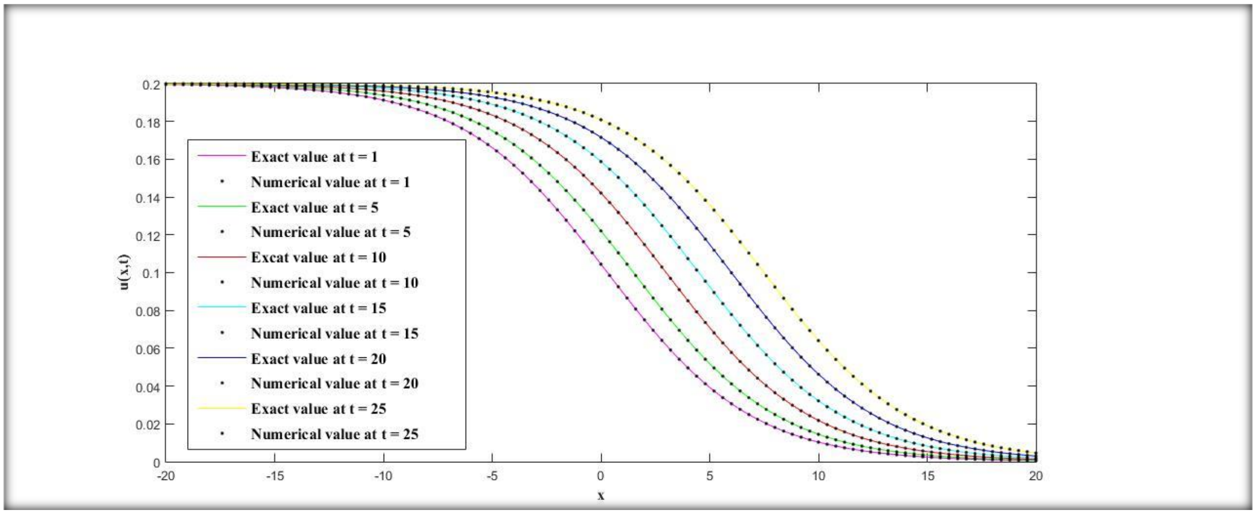

FIGURE 14.
Graphical representation of Exact and Numerical solution of u(x, t) for the parameters Δt = 0.001, N = 101, λ= 0.1 at the time levels t = 1, 5, 10, 15, 20 and 25

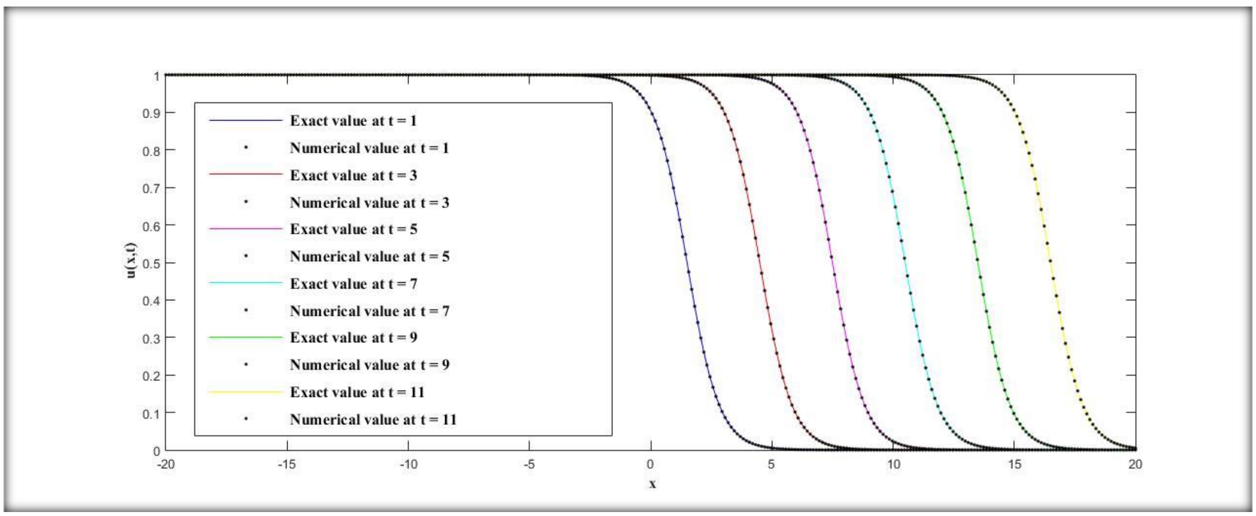

FIGURE 15.
Graphical representation of Exact and Numerical solution of u(x, t) for the parameters Δt = 0.001, N = 320, λ= 0.5 at the time levels t = 1, 3, 5, 7, 9 and 11

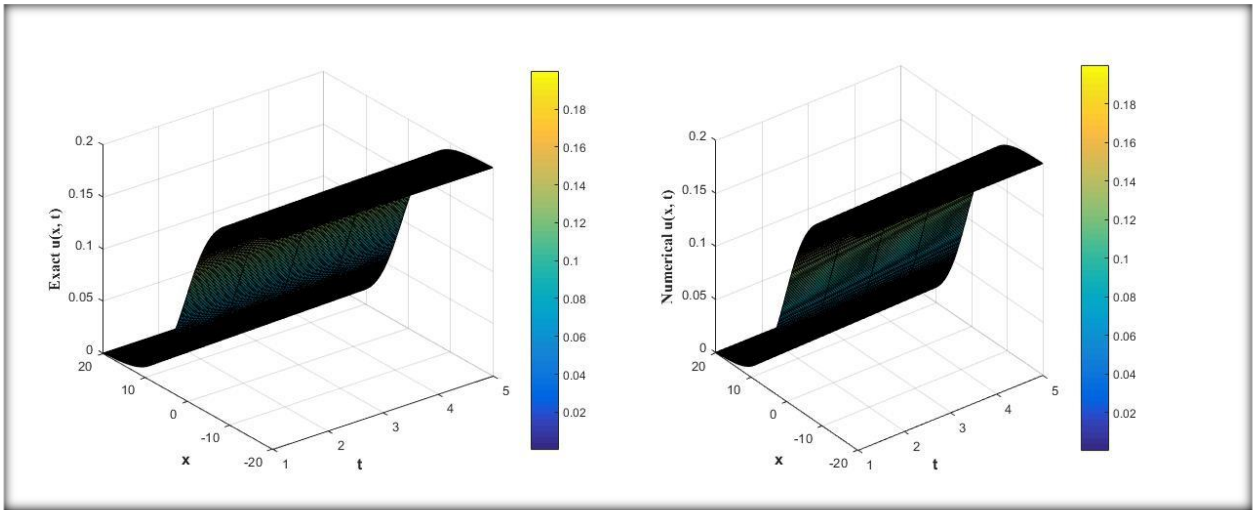

FIGURE 16.
Graphical representation of Exact and Numerical solution of u(x, t) for the parameters $\Delta t = 0.001$, $N = 320$, $\lambda = 0.1$ at the time levels t = 1, 2, 3, 4 and 5

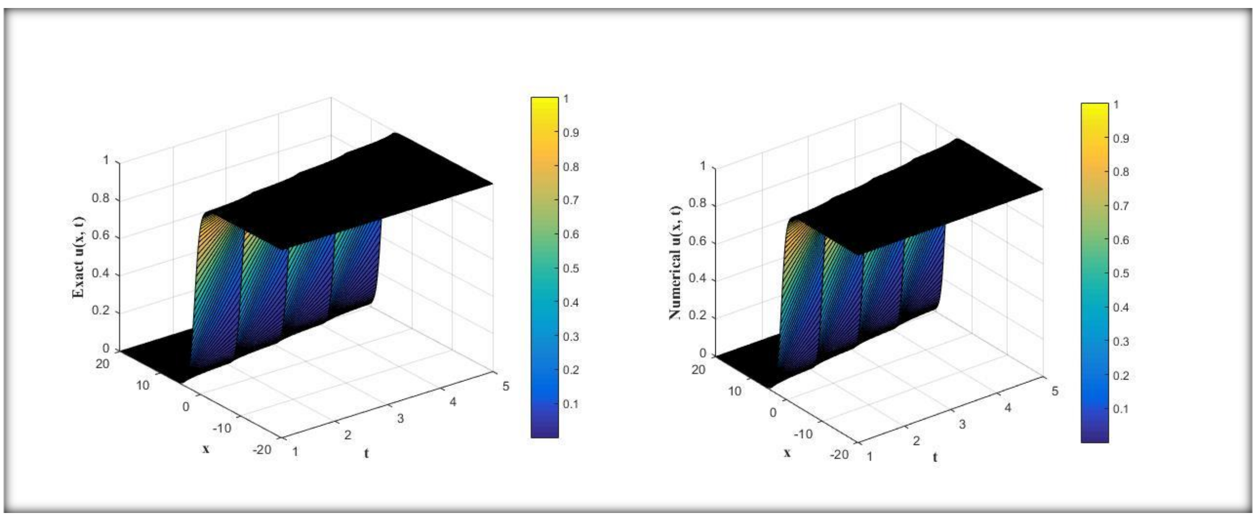

FIGURE 17.
Graphical representation of Exact and Numerical solution of u(x, t) for the parameters $\Delta t = 0.001$, $N = 320$, $\lambda = 0.5$ at the time levels t = 1, 2, 3, 4 and 5

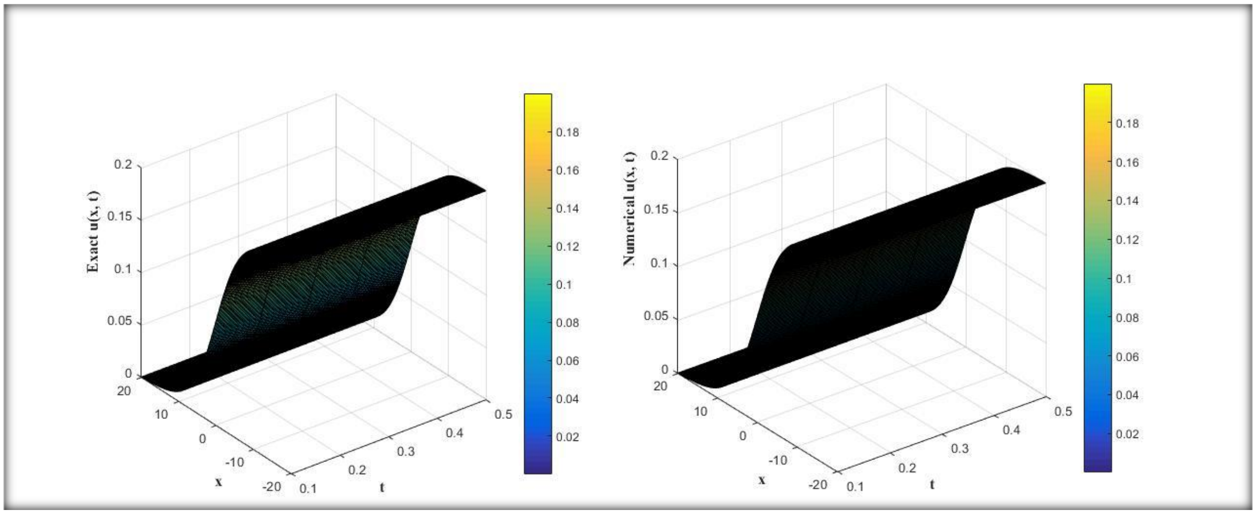

FIGURE 18.
Graphical representation of Exact and Numerical solution of u(x, t) for the parameters Δt = 0.001, N = 350, λ= 0.1 at the time levels t = 0.1, 0.2, 0.3, 0.4 and 0.5

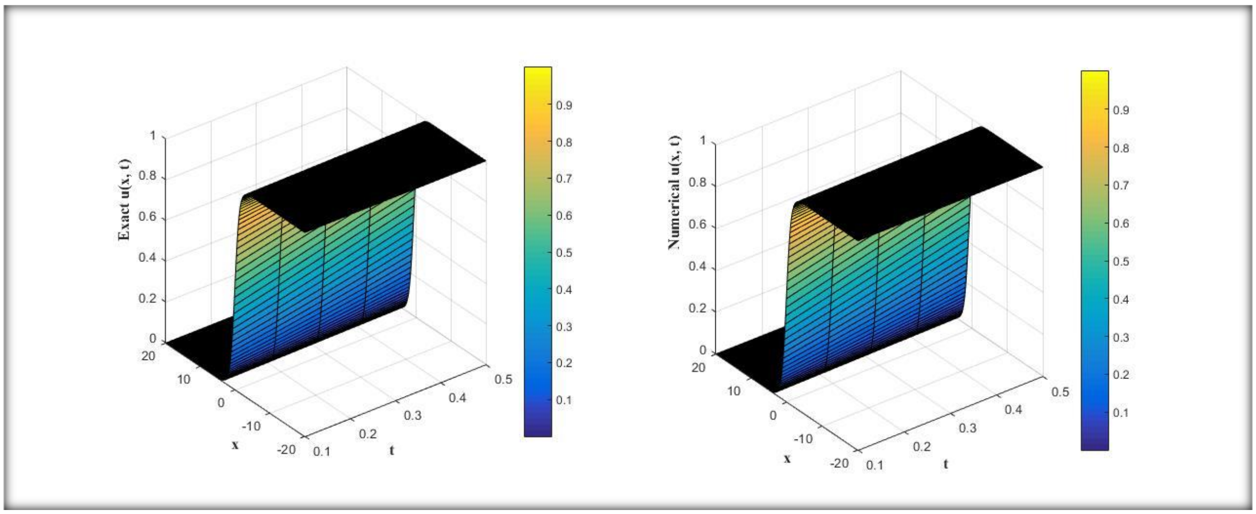

FIGURE 19.
Graphical representation of Exact and Numerical solution of u(x, t) for the parameters Δt = 0.001, N = 350, λ= 0.5 at the time levels t = 0.1, 0.2, 0.3, 0.4 and 0.5

4. Stability of proposed method

Following [52, 34, and 53], the Stability of Coupled viscous 1D Burgers' equation is discussed as follows. Coupled 1D non-linear Burgers' equation is given ahead,

$$u_t + \delta u_{xx} + \eta u u_x + \alpha (uv)_x = 0$$
$$v_t + \mu v_{xx} + \xi v v_x + \beta (uv)_x = 0$$

By discretizing the above system of 1D coupled equations, the following coupled system will be obtained,

$$\frac{du}{dt} = - \delta \sum_{j=1}^{n} a_{ij}^{(2)} u(x_j) - \eta u(x_i) \sum_{j=1}^{n} a_{ij}^{(1)} u(x_j)$$

$$- \alpha[u(x_i) \sum_{j=1}^{n} a_{ij}^{(1)} v(x_j) + v(x_i) \sum_{j=1}^{n} a_{ij}^{(1)} u(x_j)] \quad \textbf{(4.1)}$$

$$\frac{dv}{dt} = - \mu \sum_{j=1}^{n} a_{ij}^{(2)} v(x_j) - \xi v(x_i) \sum_{j=1}^{n} a_{ij}^{(1)} v(x_j) - \beta [u(x_i) \sum_{j=1}^{n} a_{ij}^{(1)} v(x_j) + v(x_i) \sum_{j=1}^{n} a_{ij}^{(1)} u(x_j)] \quad \textbf{(4.2)}$$

Discretization of u-component is given as follows

$$\frac{du}{dt} = - \delta \sum_{j=1}^{n} a_{ij}^{(2)} u(x_j) - \eta u(x_i) \sum_{j=1}^{n} a_{ij}^{(1)} u(x_j) - \alpha[u(x_i) \sum_{j=1}^{n} a_{ij}^{(1)} v(x_j) + v(x_i) \sum_{j=1}^{n} a_{ij}^{(1)} u(x_j)]$$

$$\frac{du}{dt} = - \delta[a_{i1}^{(2)} u(x_1) + \sum_{j=2}^{n-1} a_{ij}^{(2)} u(x_j) + a_{in}^{(2)} u(x_n)] - \eta u(x_i)[a_{i1}^{(1)} u(x_1) + \sum_{j=2}^{n-1} a_{ij}^{(1)} u(x_j) + a_{in}^{(1)} u(x_n)] - \alpha u(x_i) [a_{i1}^{(1)} v(x_1) + \sum_{j=2}^{n-1} a_{ij}^{(1)} v(x_j) + a_{in}^{(1)} v(x_n)] - \alpha v(x_i) [a_{i1}^{(1)} u(x_1) + \sum_{j=2}^{n-1} a_{ij}^{(1)} u(x_j) + a_{in}^{(1)} u(x_n)]$$

$$\frac{du}{dt} = - \delta[\sum_{j=2}^{n-1} a_{ij}^{(2)} u(x_j)] - \eta u(x_i)[\sum_{j=2}^{n-1} a_{ij}^{(1)} u(x_j)] - \alpha u(x_i) [\sum_{j=2}^{n-1} a_{ij}^{(1)} v(x_j)] - \alpha v(x_i) [\sum_{j=2}^{n-1} a_{ij}^{(1)} u(x_j)] + F_i \quad \textbf{(4.3)}$$

where $F_i = - \delta[a_{i1}^{(2)} u(x_1) + a_{in}^{(2)} u(x_n)] - \eta u(x_i)[a_{i1}^{(1)} u(x_1) + a_{in}^{(1)} u(x_n)] - \alpha u(x_i) [a_{i1}^{(1)} v(x_1) + a_{in}^{(1)} v(x_n)] - \alpha v(x_i) [a_{i1}^{(1)} u(x_1) + a_{in}^{(1)} u(x_n)]$

Discretization of v-component is given as follows,

$$\frac{dv}{dt} = - \mu[\sum_{j=1}^{n} a_{ij}^{(2)} v(x_j)] - \xi v(x_i) [\sum_{j=1}^{n} a_{ij}^{(1)} v(x_j)] - \beta [u(x_i) \sum_{j=1}^{n} a_{ij}^{(1)} v(x_j)] - \beta v(x_i)[\sum_{j=1}^{n} a_{ij}^{(1)} u(x_j)]$$

$$\frac{dv}{dt} = - \mu[a_{i1}^{(2)} v(x_1) + \sum_{j=2}^{n-1} a_{ij}^{(2)} v(x_j) + a_{in}^{(2)} v(x_n)] - \xi v(x_i) [a_{i1}^{(1)} v(x_1) + \sum_{j=2}^{n-1} a_{ij}^{(1)} v(x_j) + a_{in}^{(1)} v(x_n)] - \beta u(x_i) [a_{i1}^{(1)} v(x_1) + \sum_{j=2}^{n-1} a_{ij}^{(1)} v(x_j) + a_{in}^{(1)} v(x_n)] - \beta v(x_i)[a_{i1}^{(1)} u(x_1) + \sum_{j=2}^{n-1} a_{ij}^{(1)} u(x_j) + a_{in}^{(1)} u(x_n)]$$

$$\frac{dv}{dt} = - \mu[\sum_{j=2}^{n-1} a_{ij}^{(2)} v(x_j)] - \xi v(x_i) [\sum_{j=2}^{n-1} a_{ij}^{(1)} v(x_j)] - \beta u(x_i) [\sum_{j=2}^{n-1} a_{ij}^{(1)} v(x_j)] - \beta v(x_i)[\sum_{j=2}^{n-1} a_{ij}^{(1)} u(x_j)] + G_i \quad \textbf{(4.4)}$$

where $G_i = - \mu[a_{i1}^{(2)} v(x_1) + a_{in}^{(2)} v(x_n)] - \xi v(x_i) [a_{i1}^{(1)} v(x_1) + a_{in}^{(1)} v(x_n)] - \beta u(x_i) [a_{i1}^{(1)} v(x_1) + a_{in}^{(1)} v(x_n)] - \beta v(x_i)[a_{i1}^{(1)} u(x_1) + a_{in}^{(1)} u(x_n)]$

- $U = (u, v)'$
- $u = (u_2, u_3, u_4, \ldots\ldots, u_{n-1})$ and $v = (v_2, v_3, v_4, \ldots\ldots, v_{n-1})$
- $A_2 = [a_{ij}^{(2)}]$ and $A_1 = [a_{ij}^{(1)}]$
- $\frac{dU}{dt}$ = BU + H, H = (F, G)', F = $[F_i]$, G = $[G_i]$
- $B = \begin{bmatrix} A & 0 \\ 0 & A^* \end{bmatrix}$

$$\begin{bmatrix} \frac{du}{dt} \\ \frac{dv}{dt} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A^* \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} F_i \\ G_i \end{bmatrix}$$

where

$$A = -\delta A_2 - (\eta u_i + \alpha u_i + \alpha v_i)A_1$$

and

$$A^* = -\mu A_2 - (\xi u_i + \beta u_i + \beta v_i)A_1$$

Stability of this proposed scheme depends upon the stability of the matrix B. Stability of the proposed scheme is checked with the Matrix stability analysis methodpresented in following Figure 20 for different grid points.

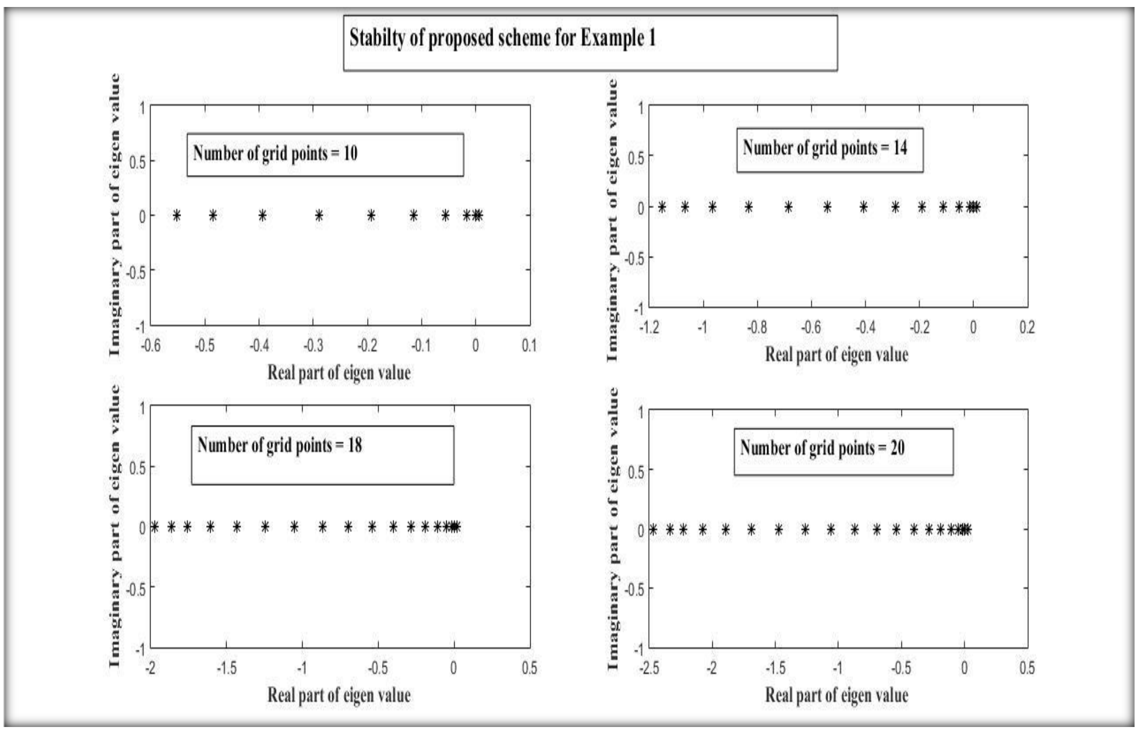

FIGURE 20: ***STABILITY OF PROPOSED SCHEME***

5. Conclusion

In the present paper, modified cubic Hyperbolic B-spline based Differential quadrature method is implemented to obtain the numerical solution of coupled 1D non-linear viscous Burgers' equation. Modified cubic Hyperbolic B-spline is used as the test function to get the weighting coefficients by using DQM; obtained set of ODEs is solved by using the SSP-RK43 scheme. The effectiveness of the proposed scheme got checked with the help of three test examples. In order to check the accurateness of the proposed scheme, obtained results are presented in the form of Tables and Figures. It has been observed that on making the comparison with previous numerical results and with exact solutions, the present scheme is acceptable. The stability of the proposed scheme is also discussed with the help of the Matrix stability analysis method, which shows that the present developed scheme is unconditionally stable.

Conflict of Interest - Authors have no conflict of interest.
Data availability statement- All data is included within the manuscript.

References

1. Esipov SE, (1995),"Coupled Burgers equations: a model of polydispersive sedimentation.", *Physical Review E*, *52*(4), 3711.
2. Kaya D, (2001),"An explicit solution of coupled viscous Burgers' equation by the decomposition method.", *International Journal of Mathematics and Mathematical Sciences*, *27*(11), 675-680.
3. Soliman A A, (2006),"The modified extended tanh-function method for solving Burgers-type equations.", *Physica A: Statistical Mechanics and its Applications*, *361*(2), 394-404.
4. AbdouM A and Soliman AA, (2005),"Variational iteration method for solving Burger's and coupled Burger's equations.", *Journal of Computational and Applied Mathematics*, *181*(2), 245-251.
5. WeiG W and Gu Y, (2002),"Conjugate filter approach for solving Burgers' equation.", *Journal of Computational and Applied Mathematics*, *149*(2), 439-456.
6. Khater A H, Temsah R S and Hassan M M, (2008), "A Chebyshev spectral collocation method for solving Burgers'-type equations.", *Journal of Computational and Applied Mathematics*, *222*(2), 333-350.
7. Dehghan M, Hamidi A andShakourifar M, (2007),"The solution of coupled Burgers' equations using Adomian–Pade technique." *Applied Mathematics and Computation*, *189*(2), 1034-1047.
8. Rashid A and Ismail A I B M, (2009),"A Fourier pseudospectral method for solving coupled viscous Burgers equations." *Computational Methods in Applied Mathematics Comput. Methods Appl. Math.*, *9*(4), 412-420.
9. Srivastava V K, Awasthi M K andTamsir M, (2013),"A fully implicit Finite-difference solution to one dimensional Coupled Nonlinear Burgers' equations.", *Int. J. Math. Sci*, *7*(4), 23.
10. Srivastava V K, Tamsir M, Awasthi M K and Singh S, (2014),"One-dimensional coupled Burgers' equation and its numerical solution by an implicit logarithmic finite-difference method." *Aip Advances*, *4*(3), 037119.
11. MittalR C and Jiwari R, (2012),"Differential quadrature method for numerical solution of coupled viscous Burgers' equations.", *International Journal for Computational Methods in Engineering Science and Mechanics*, *13*(2), 88-92.
12. Srivastava V K, Awasthi M K,Tamsir M and Singh S, (2013) "An Implicit Finite-Difference Solutiontoone-dimensionalcoupled Burgers'Equations." *Asian-European Journal of Mathematics*, *6*(04), 1350058.
13. LaiH and Ma C, (2014),"A new lattice Boltzmann model for solving the coupled viscous Burgers' equation.", *Physica A: Statistical Mechanics and its Applications*, *395*, 445-457.

14. Mittal R C and Arora G, (2011),"Numerical solution of the coupled viscous Burgers' equation.", *Communications in Nonlinear Science and Numerical Simulation*, *16*(3), 1304-1313.

15. MokhtariR,Toodar A S,Chegini N G, (2011),"Application of the Generalized Differential Quadrature Method in Solving Burgers' Equations.", *Communications in Theoretical Physics*, *56*(6), 1009.

16. Salih H M,Tawfiq L N M and Yahya Z R, (2016),"Numerical Solution of the Coupled Viscous Burgers' Equation via Cubic Trigonometric B-spline Approach.", *Math Stat*, *2*(011).

17. Raslan K R, El-Danaf T S and Ali K K, (2017),"Collocationmethodwith Quintic B-Splinemethodforsolvingcoupled Burgers' equations.", *Far East Journal of Applied Mathematics*, *96*(1), 55.

18. Liu F, Wang Y and Li S, (2018),"Barycentric interpolation collocation method for solving the coupled viscous Burgers' equations.", *International Journal of Computer Mathematics*, *95*(11), 2162-2173.

19. Li Q, Chai Z and Shi B, (2015),"A novel lattice Boltzmann model for the coupled viscous Burgers' equations.", *Applied Mathematics and Computation*, *250*, 948-957.

20. Bhatt H P, Khaliq A Q M, (2016),"Fourth-order compact schemes for the numerical simulation of coupled Burgers' equation.", *Computer Physics Communications*, *200*, 117-138.

21. Bellman R, Kashef B G and Casti J, (1972),"Differential quadrature: a technique for the rapid solution of non-linear partial differential equations,", J. Comput. Phys. 1,
40–52.

22. Quan J R, Chang C T, (1989),"New insights in solving distributed system equations by the quadrature methods-I", Comput. Chem. Eng. 13, 779–788.

23. Quan J R, Chang C T, (1989),"New insights in solving distributed system equations by the quadrature methods-II", Comput. Chem. Eng. 13, 1017–1024.

24. Shu C, Richards B E, (1990), "High resolution of natural convection in a square cavity by generalized differential quadrature", in: Proceedings of third Conference on Adv. Numer. Methods Eng. Theory Appl. Swansea, UK, 2, 978–985.

25. Shu C, (1991), "Generalized differential-integral quadrature and application to the simulation of imcompressible viscous flows including parallel computation",Ph.D. thesis, Univ. of Glasgow, UK.

26. Civalek Ö, (2004),"Application of differential quadrature (DQ) and harmonic differential quadrature (HDQ) for buckling analysis of thin isotropic plates and elastic columns.", *Engineering Structures*, *26*(2), 171-186.

27. Zhong H, (2004),"Spline-based differential quadrature for fourth order differential equations and its application to Kirchhoff plates.", *Applied Mathematical Modelling*, *28*(4), 353-366.

28. Zhong H and Lan M, (2006)," Solution of non-linear initial-value problems by the spline-based differential quadrature method.", *Journal of Sound and Vibration*, *296*(4-5), 908-918.

29. Tamsir M, Srivastava V K andJiwari R, (2016),"An algorithm based on exponential modified cubic B-spline differential quadrature method for non-linear Burgers' equation.", *Applied Mathematics and Computation*, *290*, 111-124.

30. Korkmaz A and Dağ, İ, (2011),"Shock wave simulations using sinc differential quadrature method.", *Engineering Computations*, *28*(6), 654-674.

31. KorkmazA and Dag I, (2011),"Polynomial based differential quadrature method for numerical solution of non-linear Burgers' equation.", *Journal of the Franklin Institute*, *348*(10), 2863-2875.

32. Korkmaz A, Aksoy A M and Dag I, (2011),"Quartic B-spline differential quadrature method.", *Int. J. Nonlinear Sci*, *11*(4), 403-411.

33. Arora G and Singh BK, (2013),"Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method." *Applied Mathematics and Computation*, *224*, 166-177.

34. AroraG and Joshi V, (2018),"A computational approach using modified trigonometric cubic B-spline for numerical solution of Burgers' equation in one and two dimensions.", *Alexandria Engineering Journal*, *57*(2), 1087-1098.

35. Mittal R C and Dahiya S, (2017),"Numerical simulation of three-dimensional telegraphic equation using cubic B-spline differential quadrature method.", *Applied Mathematics and Computation*, *313*, 442-452.

36. Mittal R C andJiwari R, (2012),"A differential quadrature method for numerical solutions of Burgers'-type equations." *International Journal of Numerical Methods for Heat & Fluid Flow*, *22*(7), 880-895.

37. Jiwari R, Mittal R C and Sharma K K, (2013),"A numerical scheme based on weighted average differential quadrature method for the numerical solution of Burgers' equation.", *Applied Mathematics and Computation*, *219*(12), 6680-6691.

38. Shukla H S, Tamsir M, Jiwari R and Srivastava V K, (2018),"A numerical algorithm for computation modelling of 3D non-linear wave equations based on exponential modified cubic B-spline differential quadrature method.", *International Journal of Computer Mathematics*, *95*(4), 752-766.

39. Jiwari R, Pandit S and Mittal R C, (2012),"Numerical simulation of two-dimensional sine-Gordon solitons by differential quadrature method.", *Computer Physics Communications*, *183*(3), 600-616.

40. Mittal R C andJiwari R, (2011), "Numerical solution of two-dimensional reaction–diffusion Brusselator system.", *Applied Mathematics and Computation*, *217*(12), 5404-5415.

41. Korkmaz A, (2010),"Numerical algorithms for solutions of Korteweg–de Vries equation.", *Numerical methods for partial differential equations*, *26*(6), 1504-1521.

42. Tomasiello S, (1998),"Differential quadrature method: application to initial-boundary-value problems.", *Journal of Sound and Vibration*, *218*(4), 573-585.

43. Tomasiello S, (2011), "Numerical stability of DQ solutions of wave problems", Numerical Algorithms, Vol. 57 No. 3, pp. 289-312.

44. Al KaisyAM, Esmaeel RA and Nassar MM, (2007), "Application of the differential quadrature method to the longitudinal vibration of non-uniform rods", Engineering Mechanics, Vol. 14 No. 5, pp. 303-10.

45. Hsu MH, (2009), "Differential quadrature method for solving hyperbolic heat conduction problems", Tamkang Journal of Science and Engineering, Vol. 12 No. 3, pp. 331-8.

46. NeeJ and Duan J, (1998),"Limit set of trajectories of the coupled viscous Burgers' equations.", *Applied mathematics letters*, *11*(1), 57-61.

47. Mainar E, Pena J M and Sanchez-Reyes J, (2001), "Shape preserving alternatives to the rational Bézier model", *Computer aided geometric design*, 18(1), 37-60.

48. Yu H and Guozhao W, (2005), "Constructing a quasi-Legendre basis based on the C-Bézier basis", *Progress in Natural Science*, 15(6), 559-63.

49. Wei Y W and Wang G Z, (2014), "An orthogonal basis for non-uniform algebraic-trigonometric spline space", *Applied Mathematics-A Journal of Chinese Universities*, 29(3), 273-82.

50. Nouisser O, Sbibih D and Sablonniere P, (2003), "A family of spline quasi-interpolants on the sphere", *Numerical Algorithms*, 33(1-4), 399-413.

51. Maes J and Bultheel A, (2007), "Modeling Genus Zero Closed Manifolds with Spherical Powell-Sabin B-splines", *Computer Aided Geometric Design* ***24*** 79-89.

52. Singh B K and Kumar P, (2016), "A novel approach for numerical computation of Burgers' equation in (1+ 1) and (2+ 1) dimensions", *Alexandria Engineering Journal*, *55*(4), 3331-3344.

53. Jain M K, (1983), "Numerical solution of differential equations", 2^{nd} edition., Wiley, New York, NY

54. Amirfakhrian M and Nouriani H, (2011), "Interpolation by hyperbolic B-spline functions"

55. Spiteri R J and Rutth S J, (2002), "A new class of optimal high -order strong stability preserving time discretization methods", SIAM J. NUMER. ANAL., 40(2), pp. 469-491