INTERNATIONAL JOURNAL OF MECHANICS
DOI: 10.46300/9104.2021.15.5
Volume 15, 2021

Numerical approximation of coupled 1D and 2D non-linear Burgers' equations by employing Modified Quartic Hyperbolic B-spline Differential Quadrature Method

Mamta Kapoor[1], Varun Joshi[2]
[1, 2] Department of Mathematics, Lovely Professional University, Punjab, India.

Received: October 6, 2020. Revised: March 16, 2021. Accepted: March 30, 2021. Published: April 7, 2021.

Abstract— In this paper, the numerical solution of coupled 1D and coupled 2D Burgers' equation is provided with the appropriate initial and boundary conditions, by implementing "modified quartic Hyperbolic B-spline DQM". In present method, the required weighting coefficients are computed using modified quartic Hyperbolic B-spline as a basis function. These coupled 1D and coupled 2D Burgers' equations got transformed into the set of ordinary differential equations, tackled by SSP-RK43 scheme. Efficiency of the scheme and exactness of the obtained numerical solutions is declared with the aid of 8 numerical examples. Numerical results obtained by modified quartic Hyperbolic B-spline are efficient and it is easy to implement.

Keywords—
Coupled 1D and 2D non-linear Burgers' equations, Modified Quartic Hyperbolic B-spline, Differential Quadrature Method, SSP-RK43 scheme.

I. INTRODUCTION

1.1. Coupled 1D non-linear Burgers' equation

Coupled 1D Burgers' equation is defined as:

$$U_t + \delta\, U_{xx} + \eta\, U\, U_x + \alpha\,(UV)_x = 0 \qquad (1)$$

$$V_t + \mu\, V_{xx} + \xi\, V\, V_x + \beta\,(UV)_x = 0 \qquad (2)$$

Initial conditions:

$$U(x,0) = g_1(x) \qquad (3)$$

$$V(x,0) = g_2(x) \qquad (4)$$

Boundary conditions:

$$U(x,t) = h_1(x,t) \qquad (5)$$

$$V(x,t) = h_2(x,t) \qquad (6)$$

1.2. Coupled 2D non-linear Burgers' equation

Coupled 2D Burgers' equation is given as:

$$U_t + U\, U_x + V\, U_y = \nu\,[U_{xx} + U_{yy}] \qquad (7)$$

$$V_t + U\, V_x + V\, V_y = \nu\,[V_{xx} + V_{yy}] \qquad (8)$$

Initial conditions:

$$U(x,y,0) = \psi_1(x,y) \qquad (9)$$

$$V(x,y,0) = \psi_2(x,y) \qquad (10)$$

$$D = \{(x,y): x \in [a,b], y \in [c,d]\}$$

Boundary conditions:

$$U(x,y,t) = \phi_1(x,y,t) \qquad (11)$$

$$V(x,y,t) = \phi_2(x,y,t) \qquad (12)$$

$(x, y) \in \partial D$ and $t > 0$.

Where u(x, t) is the velocity component in one dimension and $U(x,y,t)$, $V(x,y,t)$ are the velocity components in 2 dimension.

Some relevant studies regarding Burgers' equation could be find ahead. Coupled 1D Burgers' equation was derived by Esipov [1]. The system of coupled Burgers' equation is very important from the numerical aspect, as in most of the cases, analytical solutions are not available. Kaya [2] used the Adomian Decomposition Method for getting the exact solution of the coupled 1D Burgers' equation. Soliman [3] used modified extended tanh function approach. Several researchers have solved the coupled 1D Burgers' equation from the numerical point of view. Esipov [1] gave the numerical solution. Wei and Gu [4] used the conjugate filter approach. Abdou and Soliman [5] implemented the Variational Iteration Method for 1D Burgers' equation and coupled Burgers' equation. Rashid and Ismail [6] implemented Fourier pseudo spectral method. Mittal and Arora [7] employed cubic B-spline collocation approach for coupled viscous Burgers' equation. Fletcher [8] used the Hopf-Cole transformation in order to find the analytical solution of coupled 2D Burgers' equation. The numerical solution of coupled Burgers' equation is obtained by many researchers due to its demand in different fields of engineering and sciences. Some of their work is presented. Tamsir et al. [9] used the notion of extended modified cubic B-spline DQM to approximate the solution of coupled 2D Burgers' equation, in mentioned paper extended modified cubic B-spline DQM was implemented in space and strong stability preserving Runge-Kutta stages 5 and order 4 (SSP-RK 54) was employed in time, stability analysis of the method was also provided. Tamsir et al. [10] 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.

1.3. Differential Quadrature Method

DQM is a numerical discretization tool. DQM was initially proposed by Bellman and his associates [11] in 1972. DQM has widely came in to notice and emerged as a preferable method in previous decades due to its ease of application. Numerous researchers have provided the different numerical approximations based upon DQM. These different numerical regimes are mostly done by using the different test functions, like, Legendre polynomial functions, spline function [11][12], Lagrange interpolation polynomial function [13][14][15], radial basis function [16], Hermite polynomials [17], Sinc function [18][19], B-spline functions [20][21][22][23][24][25][36][37] and many others.

Present paper is divided in to different sections. In Section II, the numerical scheme (Modified Quartic Hyperbolic B-spline DQM) is elaborated completely, moreover formation of quartic Hyperbolic B-spline is provided as well as the derivative value of the quartic Hyperbolic B-spline is also evaluated. Tabular values of quartic Hyperbolic B-spline and it's derivative are calculated at the different node points. Present scheme is completely novel and has never been implemented to solve coupled 1D and coupled 2D Burgers' equations as per literature. In this work quartic Hyperbolic B-spline is developed and modified values of the mentioned Hyperbolic B-splines are implemented to solve coupled 1D and 2D coupled Burgers' equations. Results obtained by this scheme are acceptable. This work will surely help others researchers in the solution of complex non-linear partial differential equations.

II. NUMERICAL METHOD (MODIFIED QUARTIC HYPERBOLIC B-SPLINE DIFFERENTIAL QUADRATURE METHOD)

2.1 Formation of Quartic Hyperbolic B-spline

$$H_m(x) = \frac{1}{M_1} \times \begin{cases} 1.[sinh(x-x_{m-2})]^4, [x_{m-2}, x_{m-1}) \\ 2.[sinh(x-x_{m-2})]^3 sinh(x_m - x) \\ +[sinh(x-x_{m-2})]^2 sinh(x_{m+1}-x) sinh(x-x_{m-1}) \\ +sinh(x-x_{m-2}) sinh(x_{m+2}-x)[sinh(x-x_{m-1})]^2 \\ +sinh(x_{m+3}-x)[sinh(x-x_{m-1})]^3, [x_{i-1}, x_i) \\ 3.[sinh(x-x_{m-2})]^2[sinh(x_{m+1}-x)]^2 \\ +sinh(x-x_{m-2}) sinh(x_{m+2}-x) sinh(x-x_{m-1}) sinh(x_{m+1}-x) \\ +sinh(x-x_{m-2})[sinh(x_{m+2}-x)]^2 sinh(x-x_m) \\ +sinh(x_{m+3-}x)[sinh(x-x_{m-1})]^2 sinh(x_{m+1}-x) \\ +sinh(x_{m+3}-x) sinh(x-x_{m-1}) sinh(x_{m+2}-x) sinh(x-x_m) \\ +[sinh(x_{m+3}-x)]^2[sinh(x-x_m)]^2, [x_m, x_{m+1}) \\ 4.sinh(x-x_{m-2})[sinh(x_{m+2}-x)]^3 \\ +sinh(x_{m+3}-x) sinh(x-x_{m-1})[sinh(x_{m+2}-x)]^2 \\ +[sinh(x_{m+3}-x)]^2 sinh(x-x_m) sinh(x_{m+2}-x) \\ +[sinh(x_{m+3}-x)]^3 sinh(x-x_{m+1}), [x_{m+1}, x_{m+2}) \\ 5.[sinh(x_{m+3}-x)]^4, [x_{m+2}, x_{m+3}) \\ 6.0, elsewhere \end{cases}$$

Table 1: Tabular values of quartic Hyperbolic B-spline at different node points

Tabular Values	x_{m-2}	x_{m-1}	x_m	x_{m+1}	x_{m+2}	x_{m+3}
$B_m(x)$	0	b_1	b_2	b_2	b_1	0
$B_m'(x)$	0	b_3	b_4	$-b_4$	$-b_3$	0

Where $M_1 = sinh(h)sinh(2h)sinh(3h)sinh(4h)$

Derivative value of Quartic Hyperbolic B-spline is provided as,

$$H_m'(x) = \frac{1}{M_1} \times \begin{cases} (1)4[sinh(x-x_{m-2})]^3 cosh(x-x_{m-2}), [x_{m-2}, x_{m-1}) \\ (2)3[sinh(x-x_{m-2})]^2 cosh(x-x_{m-2}) sinh(x_m-x) \\ -[sinh(x-x_{m-2})]^3 cosh(x_m-x) \\ +2sinh(x-x_{m-2}) cosh(x-x_{m-2}) sinh(x_{m+1}-x) sinh(x-x_{m-1}) \\ -[sinh(x-x_{m-2})]^2 cosh(x_{m+1}-x) sinh(x-x_{m-1}) \\ +[sinh(x-x_{m-2})]^2 sinh(x_{m+1}-x) cosh(x-x_{m-1}) \\ +cosh(x-x_{m-2}) sinh(x_{m+2}-x)[sinh(x-x_{m-1})]^2 \\ -sinh(x-x_{m-2}) cosh(x_{m+2}-x)[sinh(x-x_{m-1})]^2 \\ +2sinh(x-x_{m-2}) sinh(x_{m+2}-x) sinh(x-x_{m-1}) cosh(x-x_{m-1}) \\ -cosh(x_{m+3}-x)[sinh(x-x_{m-1})]^3 \\ +3sinh(x_{m+3}-x)[sinh(x-x_{m-1})]^2 cosh(x-x_{m-1}), [x_{m-1}, x_m) \\ (3)2sinh(x-x_{m-2}) cosh(x-x_{m-2})[sinh(x_{m+1}-x)]^2 \\ -2[sinh(x-x_{m-2})]^2 sinh(x_{m+1}-x) cosh(x_{m+1}-x) \\ +cosh(x-x_{m-2}) sinh(x_{m+2}-x) sinh(x-x_{m-1}) sinh(x_{m+1}-x) \\ -sinh(x-x_{m-2}) cosh(x_{m+2}-x) sinh(x-x_{m-1}) sinh(x_{m+1}-x) \\ +sinh(x-x_{m-2}) sinh(x_{m+2}-x) cosh(x-x_{m-1}) sinh(x_{m+1}-x) \\ -sinh(x-x_{m-2}) sinh(x_{m+2}-x) sinh(x-x_{m-1}) cosh(x_{m+1}-x) \\ +cosh(x-x_{m-2})[sinh(x_{m+2}-x)]^2 sinh(x-x_m) \\ -2sinh(x-x_{m-2}) sinh(x_{m+2}-x) cosh(x_{m+2}-x) sinh(x-x_m) \\ +sinh(x-x_{m-2})[sinh(x_{m+2}-x)]^2 cosh(x-x_m) \\ -cosh(x_{m+3}-x)[sinh(x-x_{m-1})]^2 sinh(x_{m+1}-x) \\ +2sinh(x_{m+3}-x) sinh(x-x_{m-1}) cosh(x-x_{m-1}) sinh(x_{m+1}-x) \\ -sinh(x_{m+3}-x)[sinh(x-x_{m-1})]^2 cosh(x_{m+1}-x) \\ -cosh(x_{m+3}-x) sinh(x-x_{m-1}) sinh(x_{m+2}-x) sinh(x-x_m) \\ +sinh(x_{m+3}-x) cosh(x-x_{m-1}) sinh(x_{m+2}-x) sinh(x-x_m) \\ -sinh(x_{m+3}-x) sinh(x-x_{m-1}) cosh(x_{m+2}-x) sinh(x-x_m) \\ +sinh(x_{m+3}-x) sinh(x-x_{m-1}) sinh(x_{m+2}-x) cosh(x-x_m) \\ -2sinh(x_{m+3}-x) cosh(x_{m+3}-x)[sinh(x-x_m)]^2 \\ +2[sinh(x_{m+3}-x)]^2 sinh(x-x_m) cosh(x-x_m), [x_m, x_{m+1}) \\ (4)cosh(x-x_{m-2})[sinh(x_{m+2}-x)]^3 \\ -3sinh(x-x_{m-2})[sinh(x_{m+2}-x)]^2 cosh(x_{m+2}-x) \\ -cosh(x_{m+3}-x) sinh(x-x_{m-1})[sinh(x_{m+2}-x)]^2 \\ +sinh(x_{m+3}-x) cosh(x-x_{m-1})[sinh(x_{m+2}-x)]^2 \\ -2sinh(x_{m+3}-x) sinh(x-x_{m-1}) sinh(x_{m+2}-x) cosh(x_{m+2}-x) \\ -2sinh(x_{m+3}-x) cosh(x_{m+3}-x) sinh(x-x_m) sinh(x_{m+2}-x) \\ +[sinh(x_{m+3}-x)]^2 cosh(x-x_m) sinh(x_{m+2}-x) \\ -[sinh(x_{m+3}-x)]^2 sinh(x-x_m) cosh(x_{m+2}-x) \\ -3[sinh(x_{m+3}-x)]^2 cosh(x_{m+3}-x) sinh(x-x_{m+1}) \\ +[sinh(x_{m+3}-x)]^3 cosh(x-x_{m+1}), [x_{m+1}, x_{m+2}) \\ (5)-4[sinh(x_{m+3}-x)]^3 cosh(x_{m+3}-x), [x_{m+2}, x_{m+3}) \\ (6)0, elsewhere \end{cases}$$

Where

$$M_1 = sinh(h)sinh(2h)sinh(3h)sinh(4h)$$

$$b_1 = \frac{[\sinh(h)]^4}{M_1}$$

$$b_2 = \frac{2\,[\sinh(h)]^2[\sinh(2h)]^2 + \sinh(3h)\,[\sinh(h)]^3}{M_1}$$

$$b_3 = \frac{[2\,[\sinh(h)]]^2 \sinh\,(2h)}{M_1}$$

$$b_4 = \frac{2\sinh(2h)\cosh(2h)\,[\sinh(h)]^2 - \cosh(3h)\,[\sinh(h)]^3 + 3\sinh(3h)\,[\sinh(h)]^2\cosh\,(h)}{M_1}$$

Modified value of Quartic Hyperbolic B-spline is fetched from the following set of formulae [26][27][28].

$$\phi_1(x) = H_1(x) + 2\,H_0(x)$$

$$\phi_2(x) = H_2(x) - H_0(x)$$
$$\phi_j(x) = H_j(x), j = 3,4,5, \dots\dots\dots\dots, N-2 \quad (15)$$
$$\phi_{N-1}(x) = H_{N-1}(x) - H_{N+1}(x)$$
$$\phi_N(x) = H_N(x) + 2\,H_{N+1}(x)$$

2.2. Determination of weighting coefficients

Weighting coefficients can be easily obtained by implementing modified values of Quartic Hyperbolic B-spline in the discretization formula of DQM.

$$\phi_k^{(1)}(x_i) = \sum_{j=1}^{n} b_{ij}^{(1)}\,\phi_k(x_j), k = 1,2,3, \dots\dots\dots\dots, n \quad (16)$$

At grid point x_1:

$$\begin{bmatrix} \phi_1'(x_1) \\ \phi_2'(x_1) \\ \phi_3'(x_1) \\ \vdots \\ \vdots \\ \vdots \\ \phi_n'(x_1) \end{bmatrix} = \begin{bmatrix} \phi_1(x_1) & \phi_1(x_2) & \phi_1(x_3) & \cdots & \phi_1(x_n) \\ \phi_2(x_1) & \phi_2(x_2) & \phi_2(x_3) & & \phi_2(x_n) \\ & \vdots & & \ddots & \vdots \\ \phi_n(x_1) & \phi_n(x_2) & \phi_n(x_3) & \cdots & \phi_n(x_n) \end{bmatrix} \begin{bmatrix} b_{11}^{(1)} \\ b_{12}^{(1)} \\ b_{13}^{(1)} \\ \\ \\ b_{1n}^{(1)} \end{bmatrix} \quad (17)$$

At grid point x_2:

$$\begin{bmatrix} \phi_1'(x_2) \\ \phi_2'(x_2) \\ \phi_3'(x_2) \\ \vdots \\ \vdots \\ \vdots \\ \phi_n'(x_2) \end{bmatrix} = \begin{bmatrix} \phi_1(x_1) & \phi_1(x_2) & \phi_1(x_3) & \cdots & \phi_1(x_n) \\ \phi_2(x_1) & \phi_2(x_2) & \phi_2(x_3) & & \phi_2(x_n) \\ & \vdots & & \ddots & \vdots \\ \phi_n(x_1) & \phi_n(x_2) & \phi_n(x_3) & \cdots & \phi_n(x_n) \end{bmatrix} \begin{bmatrix} b_{21}^{(1)} \\ b_{22}^{(1)} \\ b_{23}^{(1)} \\ \\ \\ b_{2n}^{(1)} \end{bmatrix} \quad (18)$$

At grid point x_3:

$$\begin{bmatrix} \phi_1'(x_3) \\ \phi_2'(x_3) \\ \phi_3'(x_3) \\ \vdots \\ \vdots \\ \vdots \\ \phi_n'(x_3) \end{bmatrix} = \begin{bmatrix} \phi_1(x_1) & \phi_1(x_2) & \phi_1(x_3) & \cdots & \phi_1(x_n) \\ \phi_2(x_1) & \phi_2(x_2) & \phi_2(x_3) & & \phi_2(x_n) \\ & \vdots & & \ddots & \vdots \\ \phi_n(x_1) & \phi_n(x_2) & \phi_n(x_3) & \cdots & \phi_n(x_n) \end{bmatrix} \begin{bmatrix} b_{31}^{(1)} \\ b_{32}^{(1)} \\ b_{33}^{(1)} \\ \\ \\ b_{3n}^{(1)} \end{bmatrix} \quad (19)$$

……………………………………..
……………………………………..
……………………………………..

At grid point x_n:

$$\begin{bmatrix} \phi_1'(x_n) \\ \phi_2'(x_n) \\ \phi_3'(x_n) \\ \vdots \\ \vdots \\ \vdots \\ \phi_n'(x_n) \end{bmatrix} = \begin{bmatrix} \phi_1(x_1) & \phi_1(x_2) & \phi_1(x_3) & \cdots & \phi_1(x_n) \\ \phi_2(x_1) & \phi_2(x_2) & \phi_2(x_3) & \cdots & \phi_2(x_n) \\ & \vdots & & \ddots & \vdots \\ \phi_n(x_1) & \phi_n(x_2) & \phi_n(x_3) & \cdots & \phi_n(x_n) \end{bmatrix} \begin{bmatrix} b_{n1}^{(1)} \\ b_{n2}^{(1)} \\ b_{n3}^{(1)} \\ \\ \\ b_{nn}^{(1)} \end{bmatrix} \quad \textbf{(20)}$$

Where

$$\begin{bmatrix} \phi_1'(x_1) \\ \phi_2'(x_1) \\ \phi_3'(x_1) \\ \vdots \\ \vdots \\ \vdots \\ \phi_n'(x_1) \end{bmatrix} = \begin{bmatrix} -b_4 \\ b_3 + b_4 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix},$$

$$\begin{bmatrix} \phi_1'(x_2) \\ \phi_2'(x_2) \\ \phi_3'(x_2) \\ \vdots \\ \vdots \\ \vdots \\ \phi_n'(x_2) \end{bmatrix} = \begin{bmatrix} -2b_3 - b_4 \\ b_3 + b_4 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix},$$

$$\begin{bmatrix} \phi_1'(x_3) \\ \phi_2'(x_3) \\ \phi_3'(x_3) \\ \vdots \\ \vdots \\ \vdots \\ \phi_n'(x_3) \end{bmatrix} = \begin{bmatrix} -b_3 \\ -b_4 \\ b_4 \\ b_3 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \phi_1'(x_4) \\ \phi_2'(x_4) \\ \phi_3'(x_4) \\ \vdots \\ \vdots \\ \vdots \\ \phi_n'(x_4) \end{bmatrix} = \begin{bmatrix} 0 \\ -b_3 \\ -b_4 \\ b_4 \\ b_3 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix},$$

$$\begin{bmatrix} \phi_1'(x_5) \\ \phi_2'(x_5) \\ \phi_3'(x_5) \\ \vdots \\ \vdots \\ \vdots \\ \phi_n'(x_5) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -b_3 \\ -b_4 \\ b_4 \\ b_3 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

……………………………….
……………………………….
……………………………….

$$\begin{bmatrix} \phi_1'(x_{n-2}) \\ \phi_2'(x_{n-2}) \\ \phi_3'(x_{n-2}) \\ \vdots \\ \vdots \\ \vdots \\ \phi_n'(x_{n-2}) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ -b_3 \\ -b_4 \\ b_4 \\ b_3 \\ 0 \end{bmatrix},$$

$$\begin{bmatrix} \phi_1'(x_{n-1}) \\ \phi_2'(x_{n-1}) \\ \phi_3'(x_{n-1}) \\ \vdots \\ \vdots \\ \vdots \\ \phi_n'(x_{n-1}) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ -b_3 \\ -b_4 \\ b_4 \\ b_3 \end{bmatrix},$$

$$\begin{bmatrix} \phi_1'(x_n) \\ \phi_2'(x_n) \\ \phi_3'(x_n) \\ \vdots \\ \vdots \\ \vdots \\ \phi_n'(x_n) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ \\ -b_3 \\ b_4 + 2\,b_3 \end{bmatrix}$$

III. IMPLEMENTATION OF SCHEME

In this section developed scheme is implemented upon coupled 1D and coupled 2D equations using the differential quadrature formula. Spatial partial derivatives are dealt by the DQM formulae given as per equations (21), (22), (25), (26), (27) and (28).

3.1. Upon Coupled 1D Burgers' equation

Discretization formula for first order partial derivatives:

$$U_x^{(1)} = \sum_{j=1}^{n} w_{ij}^{(1)} U(x_j) \text{ and } V_x^{(1)} = \sum_{j=1}^{n} w_{ij}^{(1)} V(x_j) \quad \textbf{(21)}$$

Discretization formula for second order partial derivatives:

$$U_x^{(2)} = \sum_{j=1}^{n} w_{ij}^{(2)} U(x_j) \text{ and } V_x^{(2)} = \sum_{j=1}^{n} w_{ij}^{(2)} V(x_j) \quad \textbf{(22)}$$

By applying the DQM approximations (21) and (22) in coupled 1D Burgers' equations, following set of equations will be obtained.

$$\frac{dU_i}{dt} = -\delta \sum_{j=1}^{n} w_{\{ij\}}^{\{(2)\}} U(x_j) - \eta\, U_i \sum_{j=1}^{n} w_{ij}^{(1)} U(x_j) - \alpha\, (UV)_{x_i} = 0 \quad \textbf{(23)}$$

$$\frac{dV_i}{dt} = -\mu \sum_{j=1}^{n} w_{ij}^{(2)} V(x_j) - \xi\, V_i \sum_{j=1}^{n} w_{ij}^{(1)} V(x_j) - \alpha\, (UV)_{x_i} = 0 \quad \textbf{(24)}$$

3.2. Upon Coupled 2D Burgers' equation

$$\frac{\partial U(x_i,y_j,t)}{\partial x} = \sum_{k=1}^{N} w_{ik}^{(1)} U(x_k, y_j, t)$$
and
$$\frac{\partial^2 U(x_i,y_j,t)}{\partial x^2} = \sum_{k=1}^{N} w_{ik}^{(2)} U(x_k, y_j, t) \quad \textbf{(25)}$$

$$\frac{\partial U(x_i,y_j,t)}{\partial y} = \sum_{k=1}^{N} W_{jk}^{(1)} U(x_i, y_k, t)$$
and
$$\frac{\partial^2 U(x_i,y_j,t)}{\partial y^2} = \sum_{k=1}^{N} w_{jk}^{(2)} U(x_i, y_k, t) \quad \textbf{(26)}$$

$$\frac{\partial V(x_i,y_j,t)}{\partial x} = \sum_{k=1}^{N} w_{ik}^{(1)} V(x_k, y_j, t)$$
and
$$\frac{\partial^2 V(x_i,y_j,t)}{\partial x^2} = \sum_{k=1}^{N} w_{ik}^{(2)} V(x_k, y_j, t) \quad \textbf{(27)}$$

$$\frac{\partial V(x_i,y_j,t)}{\partial y} = \sum_{k=1}^{N} W_{jk}^{(1)} V(x_i, y_k, t)$$
and
$$\frac{\partial^2 V(x_i,y_j,t)}{\partial y^2} = \sum_{k=1}^{N} w_{jk}^{(2)} V(x_i, y_k, t) \quad \textbf{(28)}$$

By the means of DQM approximation formulea (25)(26)(27)(28), implementation of the scheme upon coupled 2D Burgers' equations, is given as follows:

$$\frac{dU_{ij}}{dt} = -U_{ij} \sum_{k=1}^{n} w_{ik}^{(1)} U(x_k, y_j, t) - V\_\{ij\} \sum_1^2 W_{jk}^{(1)} U(x_i, y_k, t) - \nu \left[\sum_{k=1}^{n} w_{ik}^{(2)} U(x_k, y_j, t) + \sum_{k=1}^{n} W_{jk}^{(2)} U(x_i, y_k, t)\right] \quad \textbf{(29)}$$

$$\frac{dV_{ij}}{dt} = -U_{ij} \sum_{k=1}^{n} w_{ik}^{(1)} V(x_k, y_j, t) - V\_\{ij\} \sum_1^2 W_{jk}^{(1)} V(x_i, y_k, t) - \nu \left[\sum_{k=1}^{n} w_{ik}^{(2)} V(x_k, y_j, t) + \sum_{k=1}^{n} W_{jk}^{(2)} V(x_i, y_k, t)\right] \quad \textbf{(30)}$$

The obtained system of ordinary differential equations is tackled by the means of the SSP-RK43 scheme [29][30][31]. The higher order weighting coefficients [32] are evaluated in MATLAB by the help of program.

IV. NUMERICAL EXPERIMENTS AND DISCUSSION

In present section 8 numerical examples are discussed. First three examples are associated to coupled 1D Burgers' equation and rest five examples are associated to the concept of coupled 2D Burgers' equation. L_2 and L_∞ errors norms are provided for these examples. Moreover exact solutions are matched with the numerical solutions. Via graphical representation of the results it got noticed that in all cases numerical and exact solutions are compatible. Accuracy of the scheme is verified with the aid of RMS and Relative error norms also. It is obvious with all these details the developed scheme is quite acceptable and easy to implement.

Example 1:

In this example coupled 1D Burgers' equations (1) and (2) are considered with the following exact solutions from [2], which are given as,

$$U(x,t) = a_0 - 2A \frac{(2\alpha - 1)}{(4\alpha\beta - 1)} tanh[A(x - 2At)], -10 \le x \le 10, t > 0 \quad \textbf{(31)}$$

$$V(x,t) = a_0 - \frac{(2\beta - 1)}{(2\alpha - 1)} - 2A \frac{(2\alpha - 1)}{(4\alpha\beta - 1)} tanh[A(x - 2At)], -10 \le x \le 10, t > 0 \quad \textbf{(32)}$$

Computational Domain: [-L, L] = [-10, 10]

Initial conditions:

$$U(x,0) = a_0 - 2A \frac{(2\alpha - 1)}{(4\alpha\beta - 1)} tanh[Ax], -10 \le x \le 10 \quad \textbf{(33)}$$

$$V(x,0) = a_0 - \frac{(2\beta - 1)}{(2\alpha - 1)} - 2A\frac{(2\alpha - 1)}{(4\alpha\beta - 1)} tanh[Ax], -10 \leq x \leq 10 \quad (34)$$

Boundary conditions:

$$U(-L,t) = a_0 - 2A\frac{(2\alpha - 1)}{(4\alpha\beta - 1)} tanh[A(-L - 2At)], t > 0 \quad (35)$$

$$U(L,t) = a_0 - 2A\frac{(2\alpha - 1)}{(4\alpha\beta - 1)} tanh[A(L - 2At)], t > 0 \quad (36)$$

$$V(-L,t) = a_0 - \frac{(2\beta - 1)}{(2\alpha - 1)} - 2A\frac{(2\alpha - 1)}{(4\alpha\beta - 1)} tanh[A(-L - 2At)], t > 0 \quad (37)$$

$$V(L,t) = a_0 - \frac{(2\beta - 1)}{(2\alpha - 1)} - 2A\frac{(2\alpha - 1)}{(4\alpha\beta - 1)} tanh[A(L - 2At)], t > 0 \quad (38)$$

In Table 2, L_2 and L_∞ errors are provided at time level t = 0.001, Δt = 0.0001, N = 21 at the different values of a_0, α and β. In Table 3, RMS and Relative errors for both u and v components are given at the mentioned time levels for N = 31, Δt = 0.001, a_0 = 0.005, α = 0.3 and β = 0.3. In Table 4, comparison of Exact and Numerical approximations is provided at t = 0.001 and t = 0.005 for different values of x. In Figure 1, Exact and Numerical u and v components are graphically matched at t = 0.001, 0.003 and 0.005 respectively.

Table 2: L_2 and L_∞ error at time level t = 0.001, Δt = 0.0001, N = 21, for different values of a_0, α and β

a_0	α	β	L_2 u	L_∞ u	L_2 v	L_∞ v
0.01	0.1	0.1	2.25E-05	1.06E-05	2.25E-05	1.06E-05
0.01	0.1	0.3	2.05E-05	9.72E-06	2.18E-05	1.01E-05
0.01	0.3	0.1	4.19E-05	1.97E-05	3.68E-05	1.81E-05
0.01	0.3	0.3	3.02E-05	1.43E-05	3.02E-05	1.43E-05
0.001	0.1	0.1	2.27E-07	1.07E-07	2.27E-07	1.07E-07
0.001	0.1	0.3	2.08E-07	9.79E-08	2.20E-07	1.02E-07
0.001	0.3	0.1	4.23E-07	1.98E-07	3.74E-07	1.83E-07
0.001	0.3	0.3	3.06E-07	1.44E-07	3.06E-07	1.44E-07
0.05	0.1	0.1	5.44E-04	2.59E-04	5.44E-04	2.59E-04
0.05	0.1	0.3	4.99E-04	2.38E-04	5.34E-04	2.49E-04
0.05	0.3	0.1	9.70E-04	4.58E-04	8.41E-04	4.16E-04
0.05	0.3	0.3	7.22E-04	3.43E-04	7.22E-04	3.43E-04
0.005	0.1	0.1	5.63E-06	2.66E-06	5.63E-06	2.66E-06
0.005	0.1	0.3	5.14E-06	2.43E-06	5.47E-06	2.53E-06
0.005	0.3	0.1	1.05E-05	4.94E-06	9.25E-06	4.54E-06
0.005	0.3	0.3	7.59E-06	3.57E-06	7.59E-06	3.57E-06

Table 3: Root mean square and Relative error norms for u and v components for N = 31,Δt = 0.001, a_0 = 0.005, α = 0.3, β = 0.3 at different time levels

t	RMS	RMS	Relative	Relative

	Norm u	Norm v	Error u	Error v
0.01	2.79E-05	2.79E-05	3.11E-05	3.11E-05
0.001	2.02E-06	2.02E-06	1.63E-07	1.63E-07
0.002	3.80E-06	3.80E-06	5.77E-07	5.77E-07
0.003	5.84E-06	5.84E-06	1.36E-06	1.36E-06
0.004	8.12E-06	8.12E-06	2.64E-06	2.64E-06
0.005	1.07E-05	1.07E-05	4.55E-06	4.55E-06

Table 4: Comparison of Numerical and Exact approximations of u and v components for N = 31, Δt = 0.001, a_0 = 0.005, α = 0.1, β = 0.1 at time levels = 0.001 and 0.005 mentioned, for different values of x

x	Num. u	Exact u	Num v	Exact v	Num. u	Exact u	Num v	Exact v
	t = 0.001				t = 0.005			
-8	5.12E-03	5.12E-03	5.12E-03	5.12E-03	5.12E-03	5.12E-03	5.12E-03	5.12E-03
-6	5.09E-03	5.09E-03	5.09E-03	5.09E-03	5.09E-03	5.09E-03	5.09E-03	5.09E-03
-4	5.06E-03	5.06E-03	5.06E-03	5.06E-03	5.06E-03	5.06E-03	5.06E-03	5.06E-03
-2	5.03E-03	5.03E-03	5.03E-03	5.03E-03	5.03E-03	5.03E-03	5.03E-03	5.03E-03
2	4.97E-03	4.97E-03	4.97E-03	4.97E-03	4.98E-03	4.97E-03	4.98E-03	4.97E-03
4	4.94E-03	4.94E-03	4.94E-03	4.94E-03	4.93E-03	4.94E-03	4.93E-03	4.94E-03
6	4.91E-03	4.91E-03	4.91E-03	4.91E-03	4.92E-03	4.91E-03	4.92E-03	4.91E-03
8	4.88E-03	4.88E-03	4.88E-03	4.88E-03	4.87E-03	4.88E-03	4.87E-03	4.88E-03

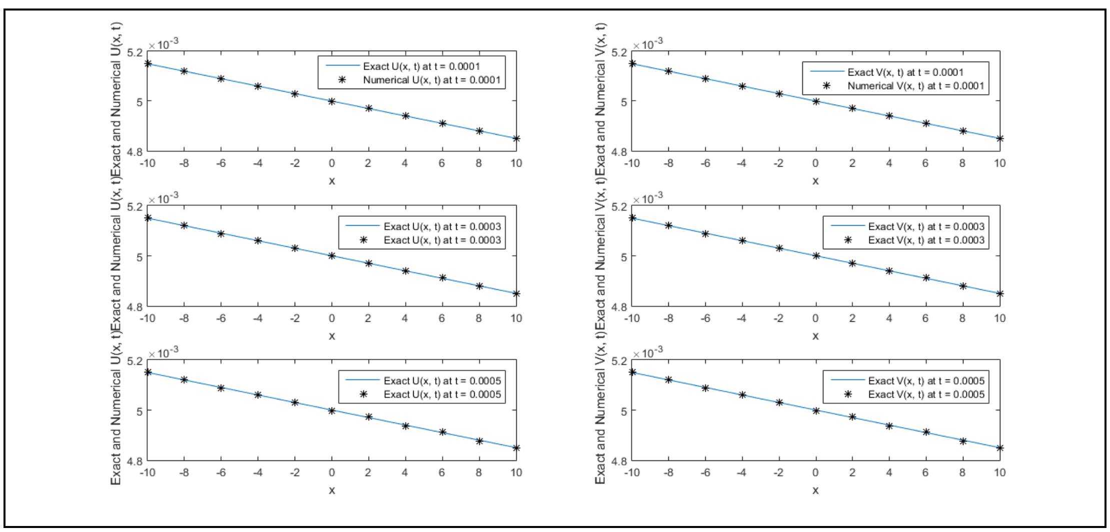

Figure 1: Plot for Exact and Numerical u and v components for N = 11, Δt = 0.0001, a_0 = 0.005, α = 0.1, β = 0.1 at time levels t = 0.0001, 0.0003 and 0.0005

Example 2:
In this example following couple 1D Burgers' equations are considered.

$$U_t - U_{xx} - 2\,U\,U_x + \frac{5}{2}\,(UV)_x = 0, -20 \leq x \leq 20, t > 0 \qquad \textbf{(39)}$$

$$V_t - V_{xx} - 2\,V\,V_x + \frac{5}{2}\,(UV)_x = 0, -20 \leq x \leq 20, t > 0 \qquad \textbf{(40)}$$

Computational Domain = [-L, L] = [-20, 20]
Exact solution is provided as [33],

$$U(x,t) = \lambda\left[1 - \tanh\left(\frac{3}{2}\lambda\,(x - 3\,\lambda\,t)\right)\right];\ -L \le x \le L;\ t > 0 \quad \textbf{(41)}$$

$$V(x,t) = \lambda\left[1 - \tanh\left(\frac{3}{2}\lambda\,(x - 3\,\lambda\,t)\right)\right];\ -L \le x \le L;\ t > 0 \quad \textbf{(42)}$$

Initial conditions:

$$U(x, 0) = \lambda\ [1 - \tanh(\tfrac{3}{2}\lambda\ x)];\ -L \le x \le L \quad \textbf{(43)}$$

$$V(x, 0) = \lambda\ [1 - \tanh(\tfrac{3}{2}\lambda\ x)];\ -L \le x \le L \quad \textbf{(44)}$$

Boundary conditions:

$$U(-L, t) = \lambda\ [1 - \tanh\tfrac{3}{2}\lambda\ (-L - 3\,\lambda\ t))];\ t > 0 \quad \textbf{(45)}$$

$$U(L, t) = \lambda\ [1 - \tanh(\tfrac{3}{2}\lambda\ (L - 3\,\lambda\ t))];\ t > 0 \quad \textbf{(46)}$$

$$V(-L, t) = \lambda\ [1 - \tanh\tfrac{3}{2}\lambda\ (-L - 3\,\lambda\ t))];\ t > 0 \quad \textbf{(47)}$$

$$V(L, t) = \lambda\ [1 - \tanh(\tfrac{3}{2}\lambda\ (L - 3\,\lambda\ t))];\ t > 0 \quad \textbf{(48)}$$

In Figure 2, comparison of Exact and Numerical u and v components is given at t = 0.0001, 0.0003 and 0.0005 for λ = 0.2. In Table 5, L2 and L_∞ error norms are provided at t = 0.0001 and t = 0.0003 for λ = 0.1 and λ = 0:2 respectively. In Table 6, Numerical and Exact components are evaluated at t =b0.001 and t = 0.005 for the different values of x. In Table 7, RMS and Relative error are provided at t = 0.001, 0.002, 0.003, 0.004 and 0.005 respectively.

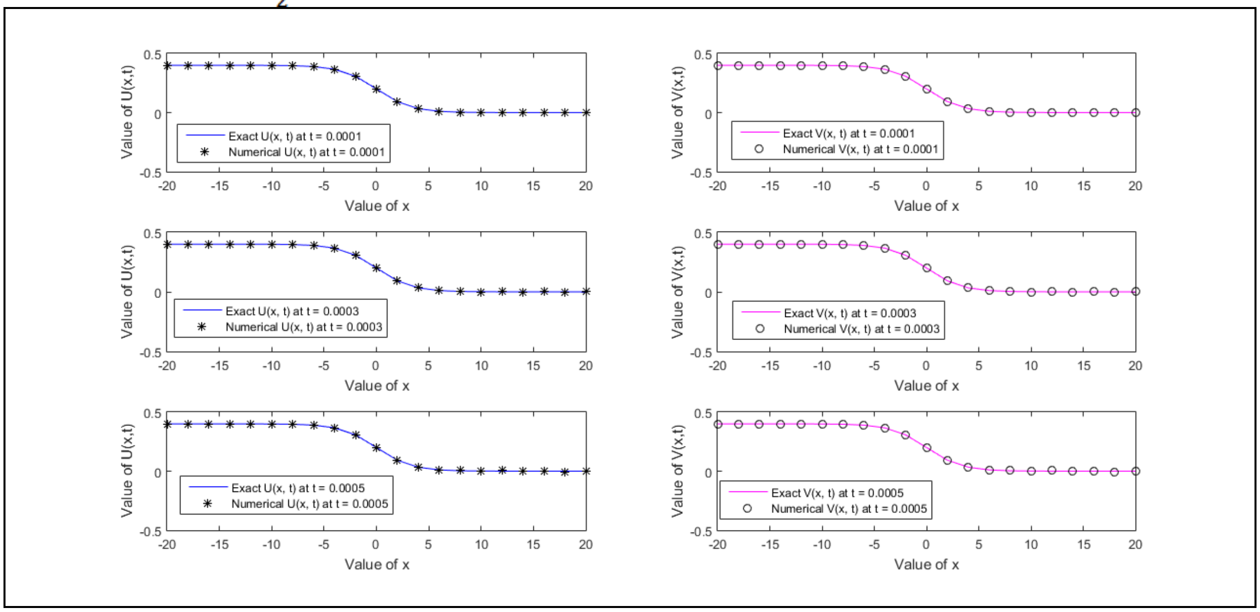

Figure 2: Comparison of Exact and Numerical u and v components for N = 21, Δt = 0.0001, λ = 0.2 at time levels t = 0.0001, 0.0003 and 0.0005

Table 5: L_2 and L_∞ error norms for N = 31, Δt = 0.0001 for λ = 0.1 and 0.2 at the time levels t = 0.0001 and t = 0.0003 respectively

	$L_2\ u$	$L_\infty\ u$	$L_2\ v$	$L_\infty\ v$
	t = 0.0001			
λ = 0.1	5.95E-04	1.59E-04	5.95E-04	1.59E-04
λ = 0.2	1.29E-03	3.20E-04	1.29E-03	3.20E-04
	$L_2\ u$	$L_\infty\ u$	$L_2\ v$	$L_\infty\ v$
	t = 0.0003			
λ = 0.1	1.80E-03	4.85E-04	1.80E-03	4.85E-04
λ = 0.2	3.90E-03	9.73E-04	3.90E-03	9.73E-04

Table 6: Comparison of Numerical and Exact u and v components for N = 31, Δt = 0.001, λ = 0.1 for different values of x at time levels t = 0.001 and t = 0.005 respectively

x	Num U	Exact U	Num V	Exact V	Num U	Exact U	Num V	Exact V
	t = 0.001				t = 0.005			
-20	2.00E-01	2.00E-01	2.00E-01	2.00E-01	2.00E-01	2.00E-01	2.00E-01	2.00E-01
-16	1.99E-01	1.98E-01	1.99E-01	1.98E-01	2.00E-01	1.98E-01	2.00E-01	1.98E-01
-12	1.94E-01	1.95E-01	1.94E-01	1.95E-01	1.94E-01	1.95E-01	1.94E-01	1.95E-01
-8	1.83E-01	1.83E-01	1.83E-01	1.83E-01	1.84E-01	1.83E-01	1.84E-01	1.83E-01
-4	1.54E-01	1.54E-01	1.54E-01	1.54E-01	1.54E-01	1.54E-01	1.54E-01	1.54E-01
0	9.93E-02	1.00E-01	9.93E-02	1.00E-01	9.68E-02	1.00E-01	9.68E-02	1.00E-01
4	4.74E-02	4.63E-02	4.74E-02	4.63E-02	5.19E-02	4.63E-02	5.19E-02	4.63E-02
8	1.52E-02	1.66E-02	1.52E-02	1.66E-02	8.66E-03	1.66E-02	8.66E-03	1.66E-02
12	6.85E-03	5.32E-03	6.85E-03	5.32E-03	1.43E-02	5.32E-03	1.43E-02	5.32E-03
16	1.88E-05	1.63E-03	1.88E-05	1.63E-03	-8.98E-03	1.63E-03	-8.98E-03	1.63E-03

Table 7: Different error norms (Root Mean Square, Relative and Average Error Norms) for N = 21, Δt = 0.001, λ = 0.1 at time levels t = 0.001, 0.002, 0.003, 0.004 and 0.005 respectively

t	RMS U	RMS V	Relative U	Relative V	AVG. Error U	AVG. Error V
0.001	2.54E-03	2.54E-03	3.83E-04	3.83E-04	1.83E-03	1.83E-03
0.002	5.57E-03	5.57E-03	1.84E-03	1.84E-03	3.97E-03	3.97E-03
0.003	9.15E-03	9.15E-03	4.97E-03	4.97E-03	6.43E-03	6.43E-03
0.004	1.34E-02	1.34E-02	1.06E-02	1.06E-02	9.26E-03	9.26E-03
0.005	1.83E-02	1.83E-02	1.98E-02	1.98E-02	1.25E-02	1.25E-02

Example 3:
In present example, considered coupled1D Burgers' equations are presented as follows:

$$U_t - U_{xx} + \eta\, U\, U_x + \alpha\,(UV)_x = 0 \quad \textbf{(49)}$$

$$V_t - V_{xx} + \xi\, V\, V_x + \beta\,(UV)_x = 0 \quad \textbf{(50)}$$

Where η, ξ, α and β all are treated as arbitrary constants.

Initial conditions:

$$U(x,0) = \begin{cases} sin(2\pi x), x \in [0, 0.5) \\ 0, x \in [0.5, 1) \end{cases} \quad \textbf{(51)}$$

and

$$V(x,0) = \begin{cases} 0, [0, 0.5) \\ -sin(2\pi x), [0.5, 1) \end{cases} \quad \textbf{(52)}$$

Boundary conditions:
In this example all boundary conditions are considered as zero.

In Figure 3, numerical approximation of u and v components is presented graphically at the different time levels for the mentioned parameters. In Figure 4, Numerical u is provided at the mentioned time levels for η = 1, ζ = 1, α =1 and β = 1. In Figure 5, Numerical profiles of v component is given at the mentioned time level for η = 1, ζ = 1, α = 1 and β = 1.

DOI: 10.46300/9104.2021.15.5

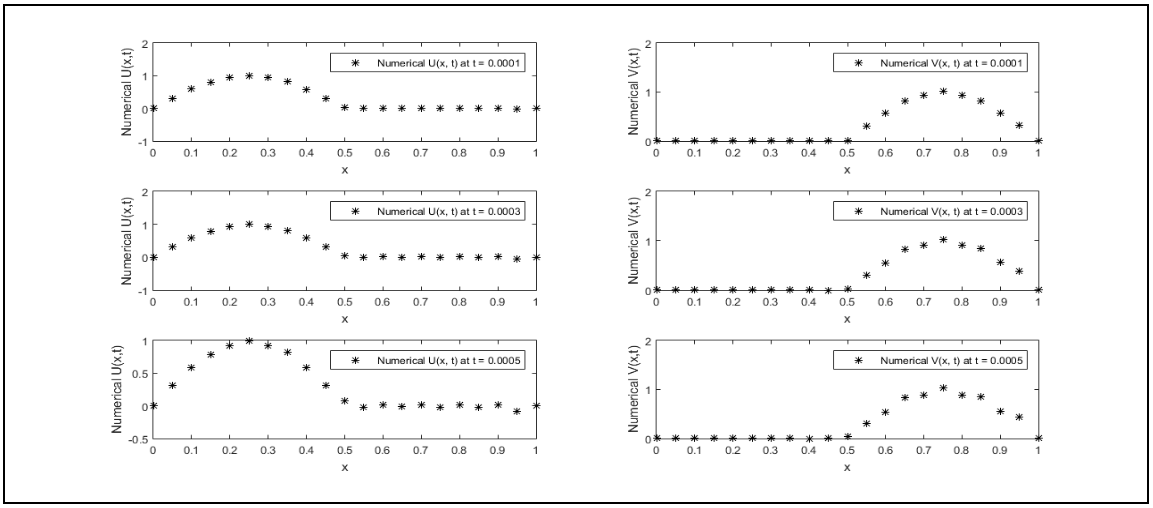

Figure 3 Numerical U(x, t) and V(x, t) for N = 21, Δt = 0.0001, η = 1, ζ = 1, α = 1 and β = 1

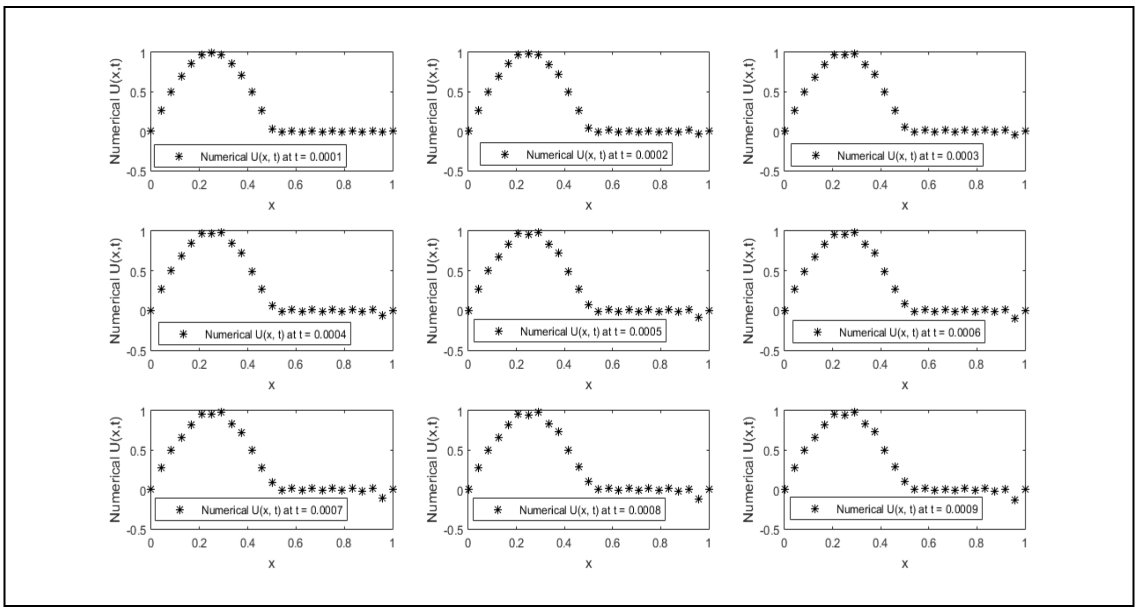

Figure 4: Numerical approximation of U(x, t) for N = 25, η = 1, ζ = 1, α = 1, β = 1, Δt = 0.0001 at different time levels

E-ISSN: 1998-4448

DOI: 10.46300/9104.2021.15.5

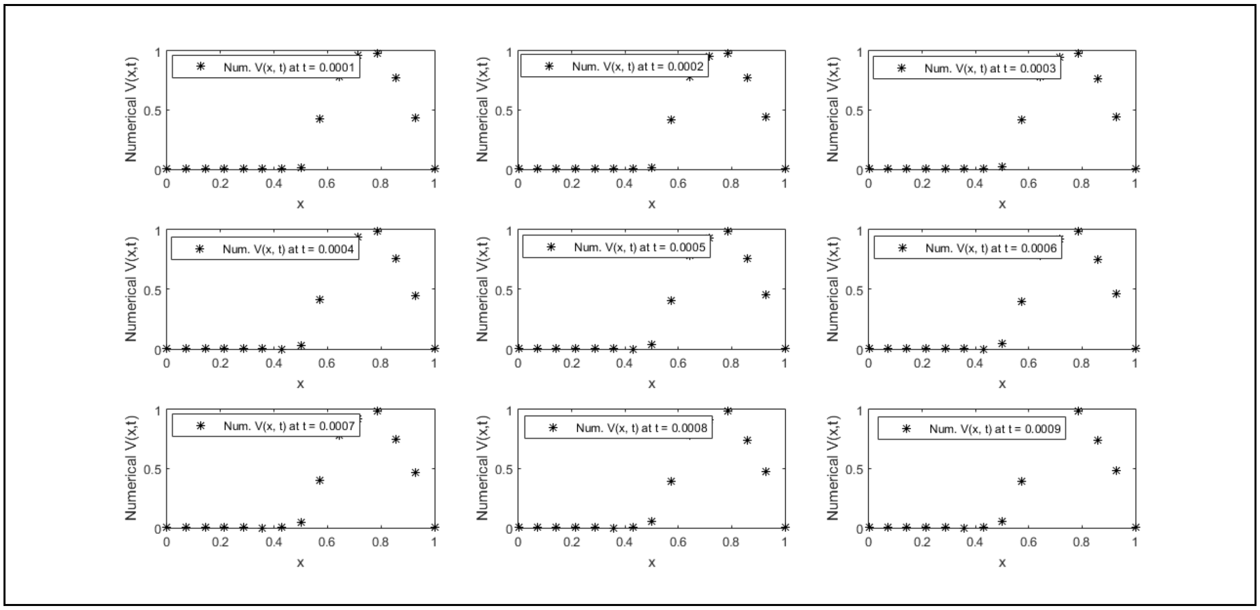

Figure 5: Numerical profile of V component for N = 10, Δt = 0.0001, η = 1, ζ = 1, α = 1, β = 1 at the different time levels

Example 4:

Considered the equations (1) and (2) with analytical solutions given by Fletcher in 1983 [8] as follows,

$$u(x,y,t) = \frac{3}{4} - \frac{1}{[4\,(1+exp((-4x+4y-t)\frac{Re}{32})]} \quad \textbf{(53)}$$

$$v(x,y,t) = \frac{3}{4} + \frac{1}{[4\,(1+\exp\,((-4x+4y-t)\frac{Re}{32})]} \quad \textbf{(54)}$$

Computational domain: [a, b] × [c, d] = [0, 1] × [0, 1]

Initial conditions:

$$u(x,y,0) = \frac{3}{4} - \frac{1}{[4\,(1+exp((-4x+4y)\frac{Re}{32})]} \quad \textbf{(55)}$$

$$v(x,y,0) = \frac{3}{4} + \frac{1}{[4\,(1+\exp\,((-4x+4y)\frac{Re}{32})]} \quad \textbf{(56)}$$

Boundary conditions:

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

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

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

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

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

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

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

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

In Figures 6, Numerical and Exact profiles of u and v components are given at t = 0.0001 for Re = 100 and Re = 200 respectively. In Figures 7, Numerical and Exact profiles of u and v components are given at t = 0.0003 for Re = 500 and Re = 1000 respectively. In Table 8, L2 and L_∞ error norms are given for different grid points at the mentioned time levels with Re = 100 and 500. In Table 9, Exact and Numerical u and v components are matched at t = 0.0001. In Table 10, RMS and Relative errors are provided at the mentioned time levels for Re = 100.

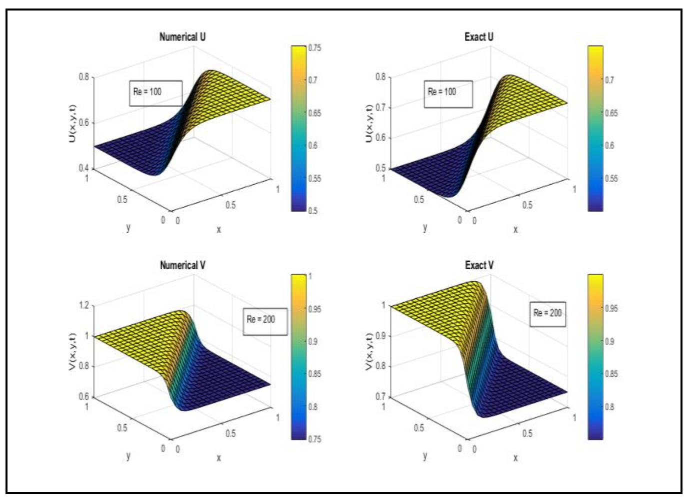

Figure 6: Numerical and Exact profile of U and V components for N = 21 × 21, Δt = 0.00001, Re = 100, 200 at time level t = 0.0001

E-ISSN: 1998-4448

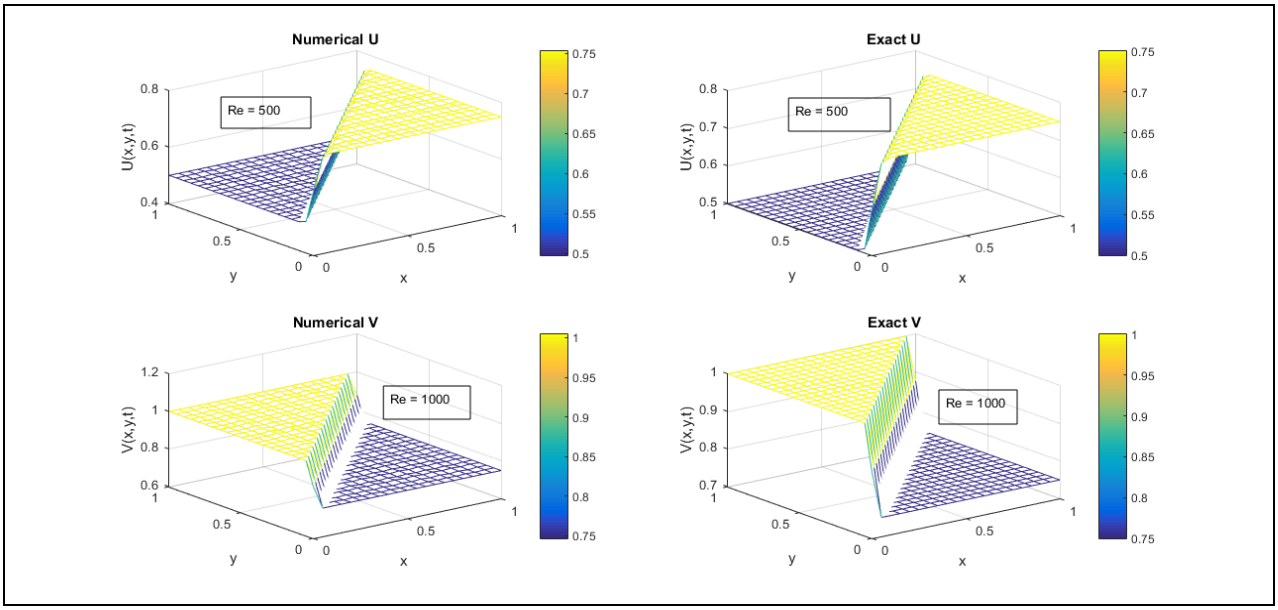

Figure 7: Numerical and Exact profile of U and V components for N = 21 × 21, Δt = 0.00001, Re = 500, 1000 at time level t = 0.0003

Table 8: Details of L_2 and L_∞ error norms at different grid points for Re = 100 and Re = 500 at the time levels t = 0.001 and t = 0.005 respectively

Grid Points	L_2 U	L_∞ U	L_2 V	L_∞ V	L_2 U	L_∞ U	L_2 V	L_∞ V
	t = 0.001, Re = 100				t = 0.005, Re = 500			
11 × 11	6.24E-04	5.50E-04	9.84E-04	8.12E-04	2.92E-03	2.83E-03	4.59E-03	3.64E-03
21 × 21	1.83E-03	1.11E-03	2.91E-03	1.62E-03	8.46E-03	5.03E-03	1.34E-02	7.05E-03
41 × 41	5.21E-03	2.34E-03	8.07E-03	3.33E-03	2.24E-02	9.67E-03	3.43E-02	1.40E-02
51 × 51	7.29E-03	3.02E-03	1.11E-02	4.26E-03	2.96E-02	1.21E-02	4.42E-02	1.75E-02
71 × 71	1.22E-02	4.65E-03	1.77E-02	6.43E-03	4.28E-02	1.68E-02	6.21E-02	2.43E-02

Table 9: Comparison of Exact and Numerical profiles of U and V components for N = 11 × 11, Δt = 0.00001, Re = 100 at time level t = 0.0001

Mesh Points	Exact U	Num. U	Exact V	Num. V
(0.1, 0.1)	0.62498	0.62486	0.87502	0.87485
(0.2,0.2)	0.62498	0.624769	0.87502	0.874713
(0.2,0.3)	0.694311	0.694473	0.805689	0.806013
(0.3,0.5)	0.73103	0.73155	0.76897	0.769639
(0.7,0.8)	0.694311	0.694356	0.805689	0.805561

Table 10: RMS and Relative error norms for U and V components for N = 11 × 11, Δt = 0.00001, Re = 100 at time levels t = 0.0001, 0.0002, 0.0003, 0.0004 and 0.0005 respectively

t	RMS U	RMS V	Relative U	Relative V
0.0001	2.38E-04	2.38E-04	2.38E-04	2.38E-04

0.0002	4.53E-04	4.53E-04	4.53E-04	4.53E-04
0.0003	6.67E-04	6.67E-04	6.67E-04	6.67E-04
0.0004	8.81E-04	8.81E-04	8.81E-04	8.81E-04
0.0005	1.09E-03	1.09E-03	1.09E-03	1.09E-03

Example 5:
In present Example considered coupled 2D Burgers' equations are having the analytical solutions as following from [34].

$$u(x,y,t) = \frac{(x+y-2xt)}{(1-2\,t^2)} \quad \textbf{(57)}$$

$$v(x,y,t) = \frac{(x-y-2yt)}{(1-2\,t^2)} \quad \textbf{(58)}$$

Computational Domain: [a, b] × [c, d] = [0, 0.5] × [0, 0.5]
Initial conditions:

$$u(x,y,0) = (x+y) \quad \textbf{(59)}$$

$$v(x,y,0) = (x-y) \quad \textbf{(60)}$$

Boundary conditions:

$$u(a,y,t) = \frac{(a+y-2at)}{(1-2\,t^2)} \quad \textbf{(61)}$$

$$u(b,y,t) = \frac{(b+y-2bt)}{(1-2\,t^2)} \quad \textbf{(62)}$$

$$u(x,c,t) = \frac{(x+c-2xt)}{(1-2\,t^2)} \quad \textbf{(63)}$$

$$u(x,d,t) = \frac{(x+d-2xt)}{(1-2\,t^2)} \quad \textbf{(64)}$$

$$v(a,y,t) = \frac{(a-y-2yt)}{(1-2\,t^2)} \quad \textbf{(65)}$$

$$v(b,y,t) = \frac{(b-y-2yt)}{(1-2\,t^2)} \quad \textbf{(66)}$$

$$v(x,c,t) = \frac{(x-c-2ct)}{(1-2\,t^2)} \quad \textbf{(67)}$$

$$v(x,d,t) = \frac{(x-d-2dt)}{(1-2\,t^2)} \quad \textbf{(68)}$$

In Figure 8, Numerical and Exact profiles of u and v components are given at t = 0.0001 for Re = 500. In Figure 9, Numerical and Exact solutions are matched at t = 0.0001 for Re = 1000. In Figure 10, Numerical and Exact profiles of both u and v components are matched at time level t = 0.0001 for Re = 1500. In Table 11, L_2 and L_∞ errors are given at the mentioned time levels for Re = 100, 500 and 1000. In Table 12, comparison of Exact and Numerical profiles is done for both components at t = 0.0001 with Re = 500.

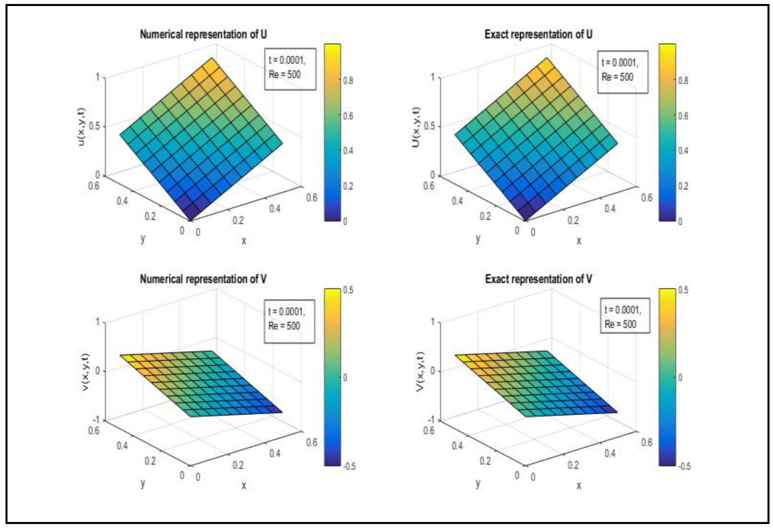

Figure 8: Numerical and Exact profiles of U and V components for N = 10 × 10, Δt = 0.00001, Re = 500 at time level t = 0.0001

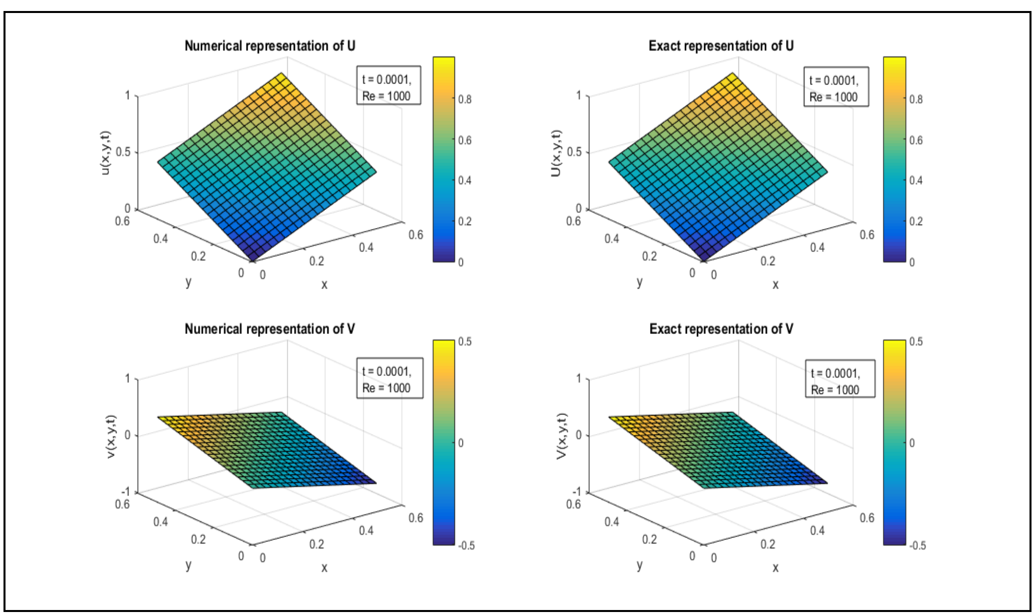

Figure 9: Numerical and Exact profiles of U and V components for N = 20 × 20, Δt = 0.00001, Re = 1000 at time level t = 0.0001

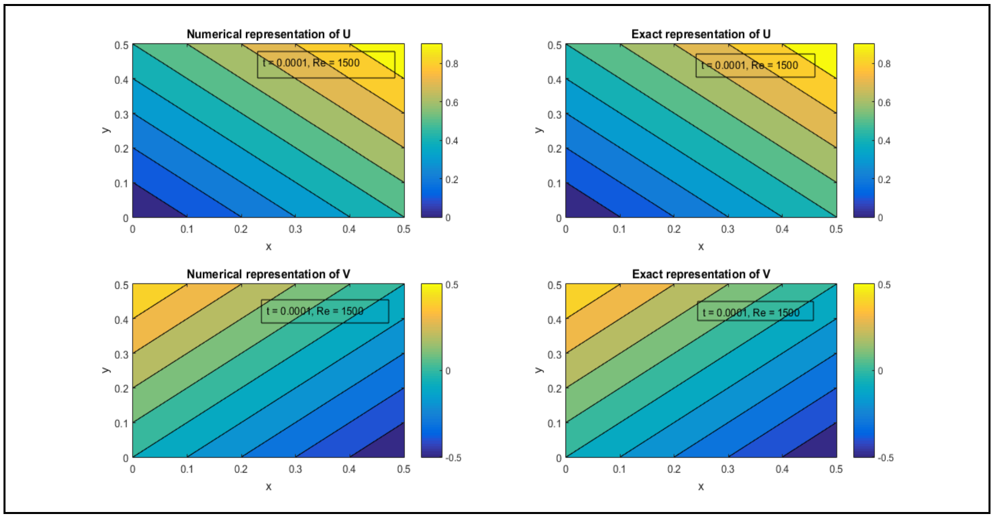

Figure 10: Numerical and Exact profiles of U and V components for N = 10 × 10, Δt = 0.00001, Re = 1500 at time level t = 0.0001

Table 11: L_2 and L_∞ error norms for N = 10 × 10, Δt = 0.00001, Re = 100, 500 and 1000 at the time levels t = 0.0001, 0.0002 and 0.0003 respectively

	L_2 U	L_∞ U	L_2 V	L_∞ V
	Re = 100			
t = 0.0001	2.66E-04	9.11E-04	1.09E-04	3.56E-04
t = 0.0002	5.05E-04	1.72E-03	2.10E-04	6.84E-04
t = 0.0003	7.43E-04	2.53E-03	3.09E-04	1.01E-03
	Re = 500			
t = 0.0001	2.36E-04	7.31E-04	1.16E-04	3.56E-04
t = 0.0002	4.48E-04	1.38E-03	2.22E-04	6.84E-04
t = 0.0003	6.59E-04	2.03E-03	3.28E-04	1.01E-03
	Re = 1000			
t = 0.0001	2.32E-04	7.09E-04	1.17E-04	3.56E-04
t = 0.0002	4.42E-04	1.34E-03	2.24E-04	6.84E-04
t = 0.0003	6.50E-04	1.96E-03	3.31E-04	1.01E-03

Table 12: Comparison of Exact and Numerical profiles of U and V components for N = 10 × 10, Δt = 0.00001, Re = 500 at the time level t = 0.0001 for different mesh points

Mesh	Exact U	Num. U	Exact V	Num. V
(0.1111, 0.111111)	0.2222	0.222186	-2.22E-05	-1.77E-05
(0.2222, 0.333333)	0.555511	0.555347	-0.11118	-0.11105
(0.5, 0.5)	0.9999	0.9999	-0.0001	-0.0001

Example 6:
In the following example Burgers' equations are given with

Computational domain: [a, b] × [c, d] = [0, 0.5] × [0, 0.5]
Initial conditions:

$$u(x, y, 0) = \sin(\pi x) + \cos(\pi y) \quad \textbf{(69)}$$
$$v(x, y, 0) = x + y \quad \textbf{(70)}$$

Boundary conditions:

$$u(0, y, t) = \cos(\pi y)$$
$$u(0.5, y, t) = 1 + \cos(\pi y)$$
$$u(x, 0, t) = 1 + \sin(\pi x)$$
$$u(x, 0.5, t) = \sin(\pi x)$$
$$v(0, y, t) = y$$
$$v(0.5, y, t) = 0.5 + y$$
$$v(x, 0, t) = x$$
$$v(x, 0.5, t) = x + 0.5$$

In Figure 11, Numerical profiles of both u and v components are given at t = 0.0001 and t = 0.0003 for Re = 1000.

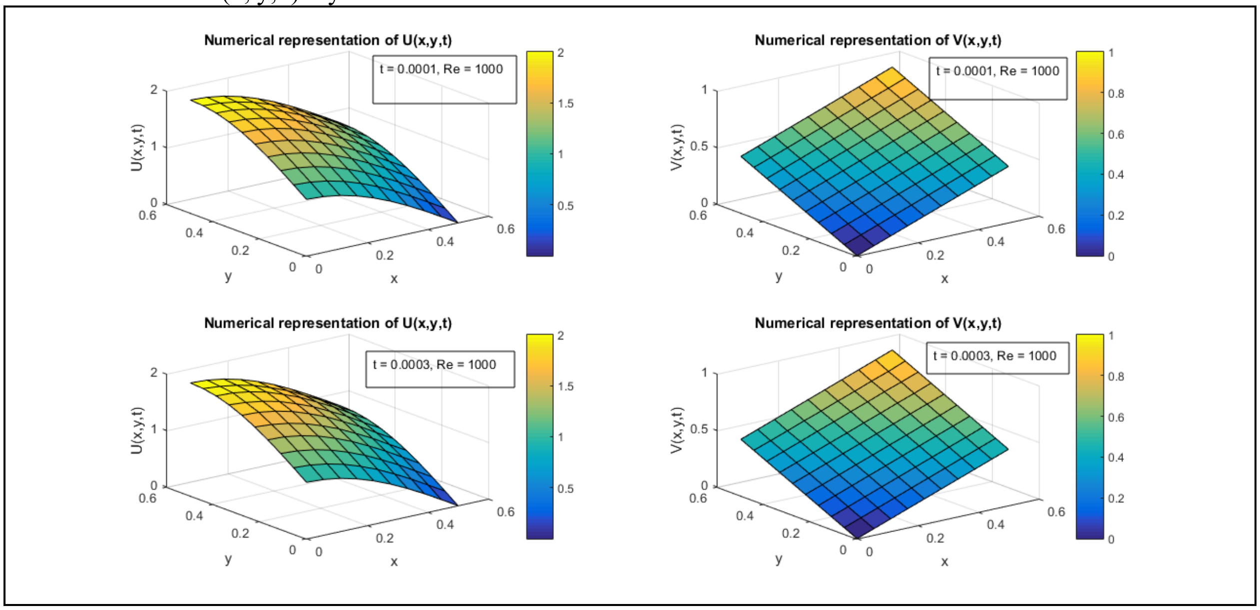

Figure 11: Numerical profiles of U and V components for N = 10, Δt = 0.000001, Re = 1000 at the time level t = 0.0001 and 0.0003 respectively

Example 7

In this example considered coupled Burgers' equations has the exact solution [35] as follows:

$$u(x, y) = \sin(\pi x)\sin(\pi y) \quad \textbf{(71)}$$
$$v(x, y) = (\sin(\pi x) + \sin(2\pi x))(\sin(\pi y) + \sin(2\pi y)) \quad \textbf{(72)}$$

Domain = $[0,1] \times [0,1]$.

In Figure 12, Numerical profiles of u and v components are given at t = 0.0001 with Re = 1000 and Re = 1500 respectively.

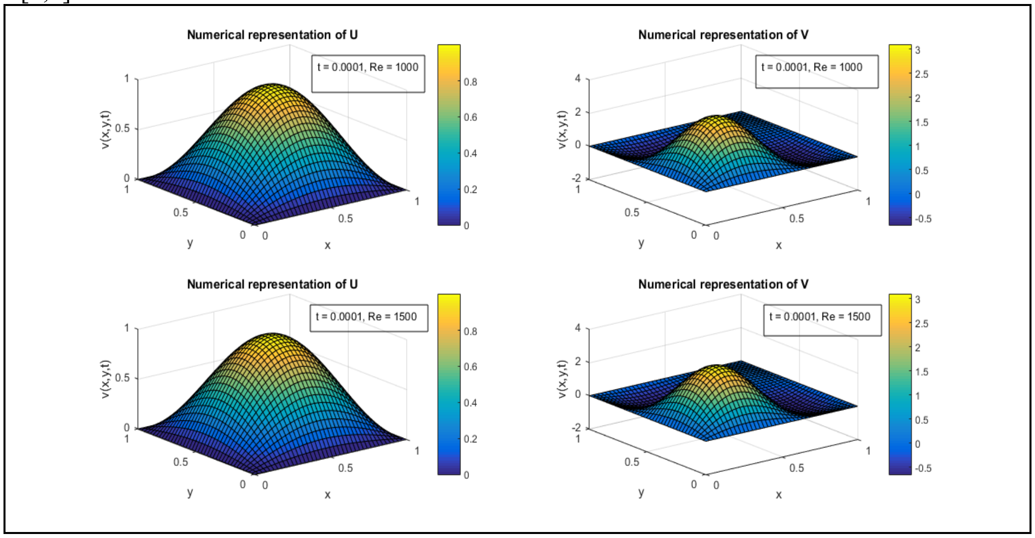

Figure 12: Numerical profile of U and V components for N = 40, Δt = 0.00001 at time level t = 0.0001 for Re = 1000 and Re = 1500 respectively

Example 8

In this example considered coupled equations are given with following Exact solutions [35].

$$u(x, y, t) = -\exp(-2\nu t)\sin(x + y) \quad \textbf{(73)}$$

$$v(x, y, t) = \exp(-2\nu t)\sin(x + y) \quad \textbf{(74)}$$

Domain = $[a, b] \times [c, d] = [-\pi, \pi] \times [-\pi, \pi]$

Initial conditions:

$$u(x, y, 0) = \sin(x + y);\ (x, y) \in [a, b] \times [c, d]$$
$$v(x, y, 0) = \sin(x + y);\ (x, y) \in [a, b] \times [c, d]$$

Boundary conditions:

$$u(a, y, t) = -\exp(-2\nu t)\sin(a + y),\ t \geq 0$$
$$u(b, y, t) = -\exp(-2\nu t)\sin(b + y),\ t \geq 0$$
$$u(x, c, t) = -\exp(-2\nu t)\sin(x + c),\ t \geq 0$$
$$u(x, d, t) = -\exp(-2\nu t)\sin(x + d),\ t \geq 0$$
$$v(a, y, t) = \exp(-2\nu t)\sin(a + y),\ t \geq 0$$
$$v(b, y, t) = \exp(-2\nu t)\sin(b + y),\ t \geq 0$$
$$v(x, c, t) = \exp(-2\nu t)\sin(x + c),\ t \geq 0$$
$$v(x, d, t) = \exp(-2\nu t)\sin(x + d),\ t \geq 0$$

In Figure 13, comparison of Numerical and Exact solutions is given graphically for both components at t = 0.0001 for Re = 500. In Figure 14, Numerical and Exact solutions are matched for u and v components at t = 0.0001 for Re = 1000. In Figure 15, a comparison of Numerical and Exact profiles of u and v components is given at t = 0.0001 for Re = 1500. In Table 13, L_2 and L_∞ errors are given at the mentioned time levels for Re = 500, 800 and 1500 respectively. In Table 14, RMS and Relative errors for both components are provided at the different time levels for Re = 1500.

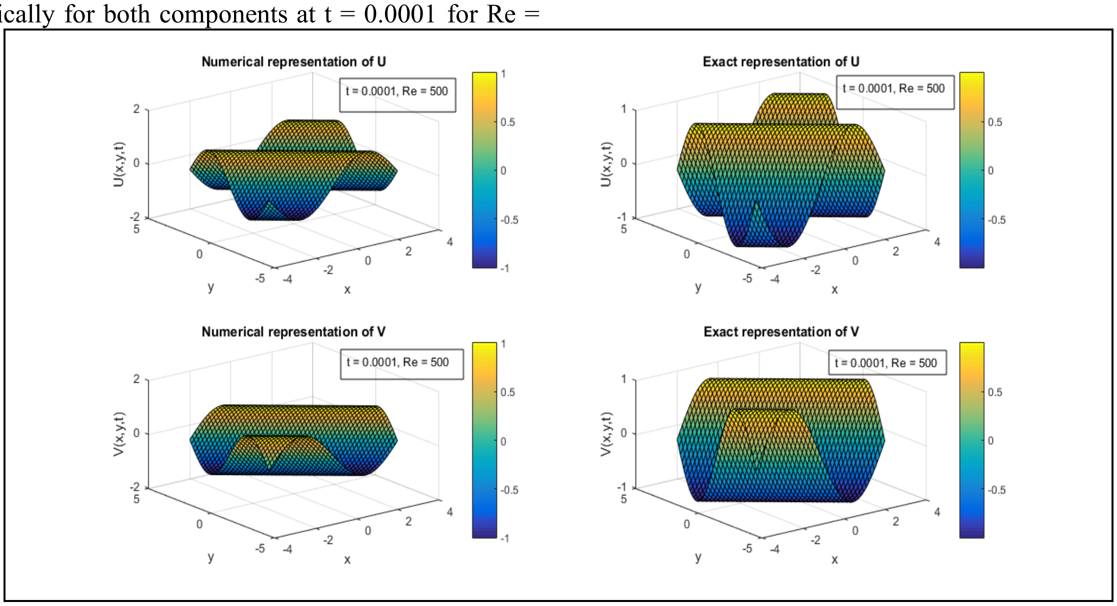

Figure 13: Comparison of Numerical and Exact profiles of U and V components for N = 50, Δt = 0.00001, Re = 500 at time level t = 0.0001

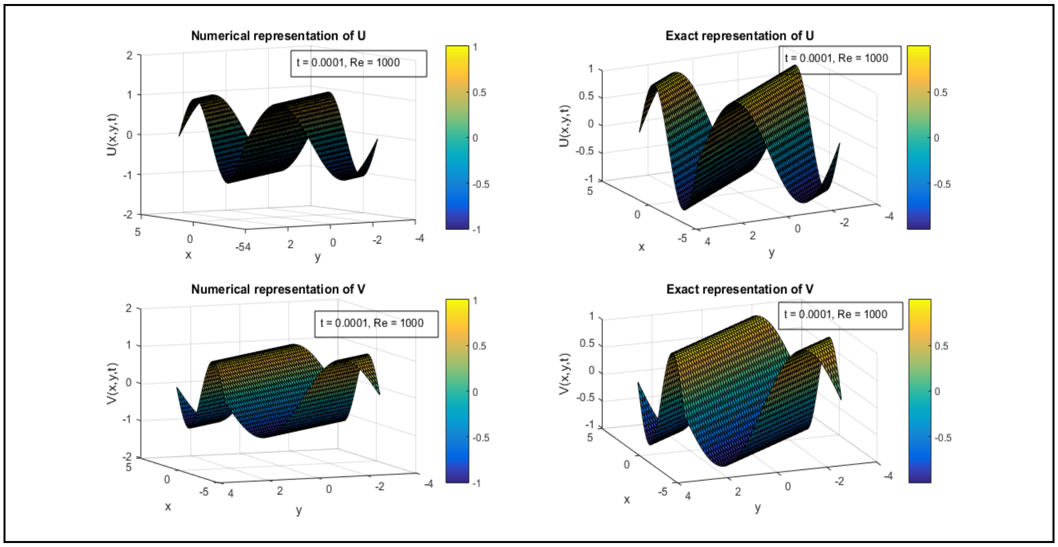

Figure 14: Comparison of Numerical and Exact profiles of U and V components for N = 50, Δt = 0.00001, Re = 1000 at time level t = 0.0001

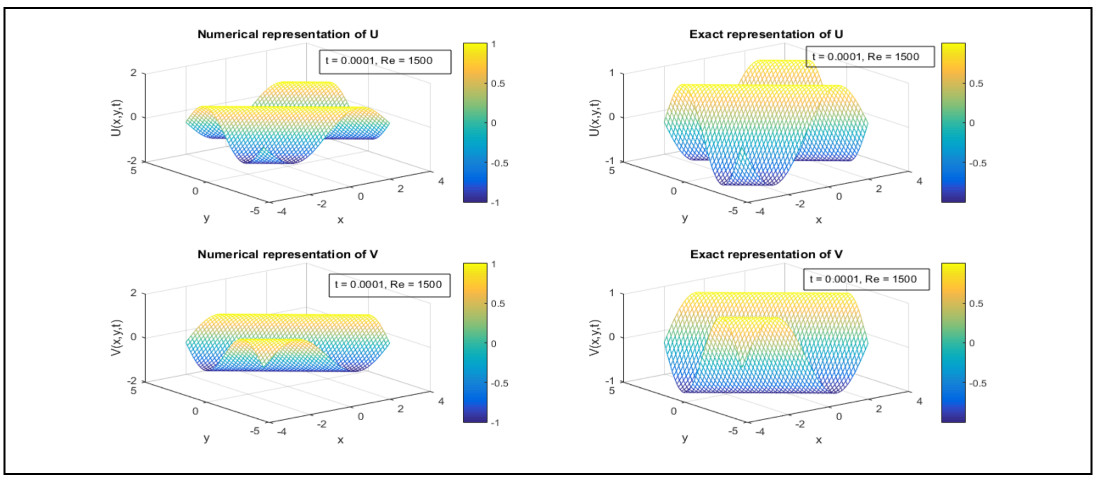

Figure 15: Comparison of Numerical and Exact profiles of U and V components for N = 50, Δt = 0.00001, Re = 1500 at time level t = 0.0001

Table 13: L_2 and L_∞ error norms for N = 10, Δt = 0.00001, Re = 500, 800 and 1500 at time levels t = 0.0001, 0.0002 and 0.0003 respectively

	L_2 U	L_∞ U	L_2 V	L_∞ V
	Re = 500			
t = 0.0001	1.16E-03	3.12E-04	1.16E-03	3.12E-04
t = 0.0002	2.22E-03	5.95E-04	2.22E-03	5.95E-04
t = 0.0003	3.28E-03	8.78E-04	3.28E-03	8.78E-04
	Re = 800			
t = 0.0001	1.16E-03	3.12E-04	1.16E-03	1.16E-03
t = 0.0002	2.22E-03	5.95E-04	2.22E-03	5.95E-04
t = 0.0003	3.27E-03	8.78E-04	3.27E-03	8.78E-04
	Re = 1500			
t = 0.0001	1.16E-03	3.12E-04	1.16E-03	3.12E-04

t = 0.0002	2.21E-03	5.95E-04	2.21E-03	5.95E-04
t = 0.0003	3.27E-03	8.78E-04	3.27E-03	3.27E-03

Table 14: RMS and Relative error norms of U and V components for N = 10, Δt = 0.00001, Re = 1500 at the mentioned time levels

t	RMS U	RMS V	Relative U	Relative V
0.0001	9.15E-05	9.15E-05	9.15E-05	9.15E-05
0.0002	1.75E-04	1.75E-04	1.75E-04	1.75E-04
0.0003	2.58E-04	2.58E-04	2.58E-04	2.58E-04

V. Conclusion

In this work, modified quartic Hyperbolic B-spline DQM is implemented upon coupled 1D and coupled 2D Burgers' equations. 8 test problems are discussed in this work. L_2 and L_∞ errors along with Root mean square and Relative errors are discussed at different parameters. Numerical approximation and Exact solutions are matched for the different values. A compatible nature of numerical and Exact values is obtained. This compatibility of the Numerical and Exact solutions declares that the results obtained from the present scheme are acceptable. This research work will help researchers in their future research work to solve some complex linear and non-linear partial differential equations. In this paper quartic Hyperbolic B-spline of fourth order is developed. But higher order Hyperbolic B-splines can also be developed to solve higher order partial differential equations, especially when the analytical solution of the partial differential equation is not available.

References

[1] S. E. Esipov, "Coupled Burgers' equations: a model of poly dispersive sedimentation", Physical Review E, vol. 52, no. 4, p. 3711, 1995.
[2] D. Kaya, "An explicit solution of coupled viscous Burgers' equation by the decomposition method", International Journal of Mathematics and Mathematical Sciences, vol. 27, 2001.
[3] A. Soliman, "The modified extended tanh-function method for solving Burgers'-type equations". Physica A: Statistical Mechanics and its Applications, vol. 361, no. 2, pp. 394–404, 2006.
[4] G. Wei and Y. Gu, "Conjugate filter approach for solving Burgers' equation", Journal of Computational and Applied Mathematics, vol. 149, no. 2, pp. 439–456, 2002.
[5] M. Abdou and A. Soliman, "Variational iteration method for solving Burger's and coupled Burger's equations", Journal of computational and Applied Mathematics, vol. 181, no. 2, pp. 245–251, 2005.
[6] A. Rashid and A. I. B. M. Ismail, "A Fourier pseudo spectral method for solving coupled viscous Burgers' equations", Computational Methods in Applied Mathematics, vol. 9, no. 4, pp. 412–420, 2009.
[7] R. C. Mittal and G. Arora, "Numerical solution of the coupled viscous Burgers' equation", Communications in Nonlinear Science and Numerical Simulation, vol. 16, no. 3, pp. 1304–1313, 2011.
[8] C. A. J. Fletcher, "Generating exact solutions of the two-dimensional Burgers' equations", IJNMF, vol. 3, pp. 213–216, 1983.
[9] 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.
[10] 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.
[11] 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.
[12] R. Bellman, B. Kashef, E. Lee, and R. Vasudevan, "Differential quadrature and splines", Computers & Mathematics with Applications, vol. 1, no. 3-4, pp. 371–376, 1975.
[13] 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.
[14] 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.
[15] 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.
[16] C. Shu and Y. Wu, "Integrated radial basis functions-based differential quadrature method and its performance", International Journal for Numerical Methods in Fluids, vol. 53, no. 6, pp. 969–984, 2007.
[17] J. Cheng, B. Wang, and S. Y. Du, "A theoretical analysis of piezoelectric/composite laminate with larger-amplitude deflection effect, part ii: Hermite differential quadrature method and application", International Journal of Solids and Structures, vol. 42, no. 24-25, pp. 6181–6201, 2005.
[18] I. Bonzani, "Solution of nonlinear evolution problems by parallelized collocation-interpolation methods", Computers & Mathematics with Applications, vol. 34, no. 12, pp. 71–79, 1997.

[19] A. Korkmaz and ˙I. Dag, "Shock wave simulations using sinc differential quadrature method", Engineering Computations, 2011.
[20] A. Bashan, "Numerical solutions of some partial differential equations with B-spline differential quadrature method", PhD, Inonu University, Malatya, Turkey, 2015.
[21] A. Bashan, S. B. G. Karakoc, and T. Geyikli, "Approximation of the KdVB equation by the quintic B-spline differential quadrature method", Kuwait journal of science, vol. 42, no. 2, 2015.
[22] A. Bashan, S. B. G. Karakoc¸, and T. Geyikli, "B-spline differential quadrature method for the modified Burgers' equation", Cankaya Universitesi Bilim ve Muhendislik Dergisi, vol. 12, no. 1, 2015.
[23] A. Bashan, Y. Ucar, N. M. Yagmurlu, and A. Esen, "Numerical solution of the complex modified Korteweg-de Vries equation by DQM", in Journal of Physics: Conference Series, vol. 766, p. 012028, 2016.
[24] S. B. G. Karakoc¸, T. Ak, and H. Zeybek, "An efficient approach to numerical study of the MRLW equation with B-spline collocation method", in Abstract and Applied Analysis, vol. 2014, Hindawi, 2014.
[25] A. Korkmaz and I. Dag, "Cubic B-spline differential quadrature methods for the advection-diffusion equation", International Journal of Numerical Methods for Heat & Fluid Flow, 2012.
[26] 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.
[27] M. Kapoor and V. Joshi, "Solution of non-linear Fisher's reaction-diffusion equation by using hyperbolic B-spline based differential quadrature method", in Journal of Physics: Conference Series, vol. 1531, p. 012064, IOP Publishing, 2020.
[28] M. Kapoor and V. Joshi, "Numerical regime uniform algebraic hyperbolic tension B-spline DQM for the solution of fisher's reaction-diffusion equation".
[29] R. Mittal and S. Dahiya, "Numerical simulation on hyperbolic diffusion equations using modified cubic B-spline differential quadrature methods", Computers & Mathematics with Applications, vol. 70, no. 5, pp. 737–749, 2015.
[30] G. Arora and V. Joshi, "Comparison of numerical solution of 1d hyperbolic telegraph equation using B-spline and trigonometric B-spline by differential quadrature method", Indian Journal of Science and Technology, vol. 9, no. 45, 2016.
[31] 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.
[32] C. Shu, Differential quadrature and its application in engineering. Springer Science & Business Media, 2012.
[33] Z. Rong-Pei, Y. Xi-Jun, and Z. Guo-Zhong, "Local discontinuous Galerkin method for solving Burgers' and coupled Burgers' equations", Chinese Physics B, vol. 20, no. 11, p. 110205, 2011.
[34] 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.
[35] 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.
[36] Y. K. DemYanovich, Y. K., I. G. Burova, T. O. Evdokimova, A. V. Lebedeva and A. G. Doronina, "Embedded spaces of hermite splines", *WSEAS Transactions on Applied and Theoretical Mechanics*, *14*, 222-234, (2019).
[37] I. G. Burova, E. G. Ivanova, V. A. Kostin, A. G. Doronina, "Trigonometric splines of the third order of approximation and interval estimation", *WSEAS Transactions on Applied and Theoretical Mechanics*, *14*, 173-183, (2019).