Numerical Approximation of Modified Burgers' equation with modified cubic NUAH B-spline Differential quadrature method

Mamta Kapoor[1], Varun Joshi[2]

[1,]Department of Mathematics, Lovely Professional University, Punjab, India.

[2]Department of Mathematics, Lovely Professional University, Punjab, India.

[1]mamtakapoor.78@yahoo.com, [2]varunjoshi20@yahoo.com

ABSTRACT

In the present paper, authors have implemented the "Modified cubic NUAH B-spline based DQM" to solve the Modified Burgers' equation from numerical aspect. Modified cubic NUAH B-spline is incorporated to discretize the spatial partial derivatives. Modified Burgers' equation is converted in to the ODE system, which later on is tackled by SSP-RK43 scheme. The numerical results of the proposed method are compared with some of the previous methods as well as with exact solution. The stability of proposed regime is verified by using "Matrix stability analysis method".

Keywords

Modified Burgers' equation, Differential quadrature method, Non-Uniform Algebraic Hyperbolic (NUAH) B-spline, Matrix stability analysis method, L_2 and L_∞ error norms.

Article Received: 10 August 2020, Revised: 25 October 2020, Accepted: 18 November 2020

Introduction

1.1. Modified Burgers' equation:

Bateman [1] gave the notion of 1D Burgers' equation. Later on it was tackled by Burgers' [2], having the following form,

$$U_t - U\,U_x + \nu U_{xx} = 0 \qquad \textbf{(1)}$$

Where ν is known as the coefficient of viscosity, x and t are subscripts, which represents space and time respectively. Burgers' equation prototype is used immensely in the area of turbulent motion, in the study of shock wave theory. Many researchers have provided theoretical as well as numerical aspect regarding Burgers' equation. Cole [3] presented relation between theory of turbulent motion and shock wave theory. Cole also presented exact solution of mentioned equation. Analytical solution of Burgers' equation is fetched, but only for some limited values of coefficient of viscosity. That is why, finding numerical approximation of Burgers' equation have attained attention of most of the researchers. Various numerical methods have been developed to fetch numerical approximation of Burgers' equation. Mittal and Singh [4] gave numerical solution of Burgers' equation. Mittal and Singh [5] presented numerical solution of periodic Burgers' equation. Ozis et al. [6] presented a FE regime for approximating Burgers' equation. Gardner et al. [7] incorporated scheme of Galerkin finite element for approximating solution of Burgers' equation. Kutluay et al. [8] solved Burgers' equation by using least-square finite element method with the aid of quadratic B-spline. Ramadan et al. [9] presented the numerical result of Burgers' equation by implementing septic B-splines.

The "Modified Burgers' equation (MBE)", is based upon Burgers' equation,

$$U_t + U^2 U_x - \nu\, U_{xx} = 0 \qquad \textbf{(2)}$$

The MBE has various non-linear concepts and it has been applied in a huge number of practical transport glitches, for example, turbulence transport, wave process in thermo-elastic phenomena, ion reflection in the quasi-perpendicular shocks, transportation and dispersion processes of the pollutants in rivers. Some of the numerical theories have been proposed to fetch numerical results of Modified Burgers' equation. Ramadan and El-Danaf [10] presented the numerical scheme for solving Modified Burgers' equation. Duan et al. [11] implemented Lattice Boltzmann method to obtain numerical solution of Modified Burgers' equation. Roshan and Bhamra [12] proposed numerical regime based upon Petrov-Galerkin method to tackle Modified Burgers' equation. Zhang et al. [13] offered notion of local discontinuous Galerkin scheme to get numerical approximation of Modified Burgers' equation. Bratsos [14] presented the numerical scheme of 4^{th} order to deal with the solution of modified Burgers' equation.

1.2. Differential quadrature method (DQM):

DQM was firstly put in to consideration by Bellman et al. [15, 16] in 1972 to solve the range of linear and non-linear partial differential equations. DQM has become very prevalent due to it's ease of application and acceptable results. The basic idea behind DQM is to attain weighting coefficients of functional values at different node points, by implementing the concept of basis functions. With aid of weighting coefficients, spatial partial derivative can be easily approximated at different node points, over entire region. Lot of work has been developed by different researchers in the area of DQM. Various DQMs are present in literature based upon a variety of test functions. Bashan et al. [17] implemented quintic B-spline DQM to solve Modified Burgers' equation. Mittal and Jiwari [18] used notion of DQM to get solution of coupled viscous Burgers' equation. Mittal and Dahiya [19] presented the concept of modified cubic B-spline DQM to tackle with approximation of Hyperbolic Diffusion equation. Jiwari [20] implemented notion of Lagrange interpolation polynomial and MCB-spline DQM for getting numerical idea of Hyperbolic PDE with Dirichlet boundary condition and Neumann boundary condition. Arora and Joshi [21] gave concept of trigonometric B-spline DQM to solve the 1D Hyperbolic Telegraph equation numerically. Arora and Joshi [22] implemented the notion of modified trigonometric cubic B-spline DQM to solve 1D and 2D Burgers' equation numerically. Mittal and Dahiya [23] presented the concept of MCB-spline DQM upon a class of non-linear viscous wave equations.

Numerical Scheme (M-C-NUAH B-spline DQM)

In present work a numerical regime named "NUAH B-spline DQM" is applied to get the numerical result of the Modified Burgers' equation. In this scheme NUAH B-spline of order four is used as the basis function in DQM, for getting the value of the required weighting coefficients. For getting improvised results, modified form [22] of the cubic NUAH B-spline is taken in to the practice. Non-Uniform Algebraic Hyperbolic (NUAH) B-spline of second order is defined as follows,

$$\phi_{m,2}(x) = \begin{cases} \dfrac{\sinh(x-x_m)}{\sinh(x_{m+1}-x_m)}, & x \in [x_m, x_{m+1}) \\ \dfrac{\sinh(x_{m+2}-x)}{\sinh(x_{m+2}-x_{m+1})}, & x \in [x_{m+1}, x_{m+2}) \end{cases}$$

(3)

Where,

$$\phi_{m,j}(x) = \int_{-\infty}^{x} (\delta_{m,j-1}\, \phi_{m,j-1}(s)ds - \delta_{m+1,j-1}\, \phi_{m+1,m-1}(s)ds)$$

$, 3 \le j \le k$ **(4)**

and

$$\delta_{m,j} = \frac{1}{\int_{-\infty}^{\infty}\phi_{m,j}(s)ds}, \quad \phi_{m,j}(s) \neq 0 \qquad \textbf{(5)}$$

By using NUAH B-spline of second order and the given recurrence relation NUAH B-spline of order 3 can be fetched.

$$\phi_{m,3}(x) = \int_{-\infty}^{x} (\delta_{m,2}\, \phi_{m,2}(s)\,ds - \delta_{m+1,2}\, \phi_{m+1,2}(s)ds) \qquad \textbf{(6)}$$

$$\phi_{m,3}(\text{x}) = \begin{cases} \dfrac{\delta_{m,2}[\cosh(x-x_m)-1]}{\sinh(x_{i+1}-x_i)}, & [x_m, x_{m+1}) \\ 1 - \dfrac{[\delta_{m,2}\{\cosh(x_{m+2}-x)-1\}]}{\sinh(x_{m+2}-x_{m+1})} - \dfrac{[\delta_{m+1,2}\,\{\cosh(x-x_{m+1})-1\}]}{\sinh(x_{m+2}-x_{m+1})}, & [x_{m+1}, x_{m+2}) \\ \dfrac{\delta_{m+1,2}[\cosh(x_{m+3}-x)-1]}{\sinh(x_{m+3}-x_{m+2})}, & [x_{m+2}, x_{m+3}) \\ 0, & elsewhere \end{cases} \qquad \textbf{(7)}$$

Similarly from the recurrence relation provided for NUAH B-spline and NUAH B-spline of third order we can get the NUAH B-spline of fourth order

$$\phi_{m,4}(x) = \int_{-\infty}^{x} [\delta_{m,3}\, NUAH_{m,3}(s)ds - \delta_{m+1,3}\, NUAH_{m+1,3}\,(s)ds] \qquad \textbf{(8)}$$

$$\phi_{m,4}(x) = \begin{cases} (1)\ \delta_{m-2,3}\,\delta_{m-2,2}\dfrac{[\sinh(x-x_{m-2})-(x-x_{m-2})]}{\sinh(x_{m-1}-x_{m-2})}, \quad \boldsymbol{x} \in [\boldsymbol{x}_{m-2}, \boldsymbol{x}_{m-1}) \\ (2)\ \delta_{m-2,3}[\dfrac{\delta_{m-2,2}\{\sinh(x_{m-1}-x_{m-2})-(x_{m-1}-x_{m-2})\}}{\sinh(x_{m-1}-x_{m-2})} + (x-x_{m-1}) + \dfrac{\delta_{m-2,2}\{\sinh(x_m-x)-\sinh(x_m-x_{m-1})+(x-x_{m-1})\}}{\sinh(x_m-x_{m-1})} \\ \qquad - \dfrac{\delta_{m-1,2}\{\sinh(x-x_{m-1})-(x-x_{m-1})\}}{\sinh(x_m-x_{m-1})}] \\ \qquad \delta_{m-1,3}\dfrac{\delta_{m-1,2}[\sinh(x-x_{m-1})-(x-x_{m-1})]}{\sinh(x_m-x_{m-1})}, \quad \boldsymbol{x} \in [\boldsymbol{x}_{m-1}, \boldsymbol{x}_m) \\ (3)\ 1 - \delta_{m-2,3}\delta_{m-2,2}\dfrac{[\sinh(x_{m+1}-x)-(x_{m+1}-x)]}{\sinh(x_{m+1}-x_m)} \\ \qquad - \delta_{m-1,3}[\dfrac{\delta_{m-1,2}\{\sinh(x_m-x_{m-1})-(x_m-x_{m-1})\}}{\sinh(x_m-x_{m-1})} + (x-x_m) + \\ \qquad \dfrac{\delta_{m-1,2}\,\{\sinh(x_{m+1}-x)-\sinh(x_{m+1}-x_m)+(x-x_m)\}}{\sinh(x_{m+1}-x_m)} - \\ \qquad \dfrac{\delta_{m,2}\{\sinh(x-x_m)-(x-x_m)\}}{\sinh(x_{m+1}-x_m)}], \quad \boldsymbol{x} \in [\boldsymbol{x}_m, \boldsymbol{x}_{m+1}) \\ (4)\ \dfrac{\delta_{m-1,3}\,\delta_{m,2}\,[\sinh(x_{m+2}-x)-(x_{m+2}-x)]}{\sinh(x_{m+2}-x_{m+1})}, \quad \boldsymbol{x} \in [\boldsymbol{x}_{m+1}, \boldsymbol{x}_{m+2}) \\ (5)\ 0, elsewhere \end{cases}$$

(9)

By DQM, r^{th} partial derivatives of U can be defined as follows

$$U_x^{(r)} = \sum_{j=1}^{n} p_{ij}^{(r)}\; U(x_j) \qquad \textbf{(10)}$$

By using $r = 1$ in above equation 1st order partial derivatives of U w.r.t. x can be approximated as,

$$U_x^{(1)} = \sum_{j=1}^{n} p_{ij}^{(1)}\; U(x_j) \qquad \textbf{(11)}$$

and the 2nd order partial derivatives of U w.r.t. x can be expressed as,

$$U_x^{(2)} = \sum_{j=1}^{n} p_{ij}^{(2)}\; U(x_j) \qquad \textbf{(12)}$$

DQM approximation of the derivatives (partial in nature) can be easily fetched by the subsequent formula, where $\psi_k(x_i)$ is considered as the modified NUAH B-spline of order 4.

$$\psi_k^{(1)}(x_i) = \sum_{j=1}^{n} a_{ij}^{(1)}\; \psi_k(x_j) \qquad \textbf{(13)}$$

By using the values given in equation (9), equation (13) will be converted in to the subsequent structure of equations,

$$A \quad \vec{a}^{(1)}[i] \;=\; \vec{R}[i] \quad \text{, where } i = 1, 2, , 3, \dots\dots, n \qquad \textbf{(14)}$$

Where i = 1, 2, 3, ….., n and k = 1, 2, 3, ……, n

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

$$\vec{a}^{(1)}[1] = \begin{pmatrix} a_{1,1}^{(1)} \\ a_{1,2}^{(1)} \\ a_{1,3}^{(1)} \\ \vdots \\ \vdots \\ a_{1,N-1}^{(1)} \\ a_{1,N}^{(1)} \end{pmatrix} \text{ and } \vec{R}[1] = \begin{pmatrix} \psi_1'(x_1) \\ \psi_2'(x_1) \\ \psi_3'(x_1) \\ \vdots \\ \vdots \\ \psi_{n-1}'(x_1) \\ \psi_n'(x_1) \end{pmatrix} = \begin{pmatrix} 2F \\ D-F \\ 0 \\ \vdots \\ \vdots \\ . \\ 0 \end{pmatrix}$$

$$\vec{a}^{(1)}[2] = \begin{pmatrix} a_{21}^{(1)} \\ a_{22}^{(1)} \\ a_{23}^{(1)} \\ \vdots \\ \vdots \\ a_{2N-1}^{(1)} \\ a_{2N}^{(1)} \end{pmatrix} \text{ and } \vec{R}[2] = \begin{pmatrix} \psi_1'(x_2) \\ \psi_2'(x_2) \\ \psi_3'(x_2) \\ \vdots \\ \vdots \\ \psi_{n-1}'(x_2) \\ \psi_n'(x_2) \end{pmatrix} = \begin{pmatrix} F \\ 0 \\ D \\ 0 \\ 0 \\ \vdots \\ . \\ 0 \end{pmatrix}$$

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

$$\vec{a}^{(1)}[n] = \begin{pmatrix} a_{n,1}^{(1)} \\ a_{n,2}^{(1)} \\ a_{n,3}^{(1)} \\ \vdots \\ \vdots \\ a_{n,n-1}^{(1)} \\ a_{n,n}^{(1)} \end{pmatrix} \text{ and } \vec{R}[n] = \begin{pmatrix} \psi_1'(x_n) \\ \psi_2'(x_n) \\ \psi_3'(x_n) \\ \vdots \\ \vdots \\ \psi_{n-1}'(x_n) \\ \psi_n'(x_n) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ . \\ \vdots \\ . \\ 0 \\ F-D \\ 2D \end{pmatrix}$$

2.1. Implementation of Numerical scheme:

By implementing the formulae of partial derivative approximation, Modified Burgers' equation will be of the following form,

$$\frac{dU_i}{dt} = -U_i^2 \sum_{j=1}^{n} p_{ij}^{(1)}\; U(x_j) + \nu \sum_{j=1}^{n} p_{ij}^{(2)}\, U(x_j) \qquad \textbf{(15)}$$

The above-mentioned arrangement of ordinary differential equations is tackled with the aid of SSP-RK43 scheme [19]. For verifying the accuracy of the proposed scheme, L_2 and L_∞ error norms have been implemented.

Formulae of error norms:

$$L_2 = [h \sum_{j=1}^{n} (U_{Exact} - U_{Numerical})^2]^{\frac{1}{2}} \qquad \textbf{(16)}$$

$$L_\infty = max\; |U_{Exact} - U_{Numerical}| \qquad \textbf{(17)}$$

Formula for Order of Convergence:

$$\text{ROC} = \frac{\ln\left(\frac{E(N_2)}{E(N_1)}\right)}{\ln\left(\frac{N_1}{N_2}\right)} \qquad \textbf{(18)}$$

Numerical Instances and Discussion:

In present segment one numerical example is discussed in detail with the aid of numerical computation. For checking the precision of proposed strategy, formulae of L_2 and L_∞ are used as well as order of convergence is also provided.

Example 1:

$$U(x, t) = \frac{\frac{x}{t}}{1+\frac{\sqrt{t}}{c_0} exp(\frac{x^2}{4\,\nu\, t})}$$

Where c_0 is considered as constant between 0 and 1. We have considered $c_0 = \frac{1}{2}$ for numerical computation.

Initial condition (I.C.):

$$U(x, 1) = \frac{x}{1+\frac{1}{c_0} exp(\frac{x^2}{4\,\nu})}$$

Boundary conditions (B.C.):

U(0, t) = 0 and U(1, t) = 0

In Table 1, L_2 and L_∞ errors are given at time levels $t = 2$ and $t = 2.5$ respectively, for the different grid points for the mentioned parameters. In Table 2, numerical and Exact solutions are compared at time levels $t = 1.5$ and $t = 1.8$ respectively for changed values of x at the mentioned parameters. These values are in good agreement. In Table 3, assessment of L_2 and L_∞ errors is done with [24] and [9] respectively. It can be observed that the present values of both error norms are much improved than previous values. In Table 3, order of convergence is mentioned at the time levels $t = 2$ and $t = 2.5$ respectively. In Figure 1, pictorial depiction of exact and numerical solutions is given at time levels $t = 1.5, 2, 2.5$ and 3 respectively. both type of solutions are exactly matched. In Figure 2, Exact and Numerical results are provided graphically at the time levels $t = 1.1, 1.2, 1.3, 1.4$ and 1.5 respectively. In Figure 3, Exact and Numerical results are matched at the time levels $t = 2.1, 2.2, 2.3$ and 2.4 respectively for the mentioned parameters. In Figure 4, graph between the value of x and Absolute error is provided at the time levels $t = 2.5, 2.6, 2.7$ and 2.8 respectively. In Table 5, a detailed comparison of the present results regarding L_2 and L_∞ errors is provided. Current results are better than some of the preceding results.

Table 1: L_2 and L_∞ error norms for different grid points at time levels $t = 2$ and $t = 2.5$ respectively with $\Delta t = 0.001, \nu = 0.001$ and $c_0 = \frac{1}{2}$

N	L_2 Error $t = 2$	L_∞ Error $t = 2$	L_2 Error $t = 2.5$	L_∞ Error $t = 2.5$
51	7.0670e-05	2.5703e-04	6.6182e-05	2.4201e-04
101	6.8411e-05	2.5810e-04	6.5217e-05	2.4327e-04
151	6.8216e-05	2.6020e-04	6.5143e-05	2.4530e-04
201	6.8170e-05	2.6094e-04	6.5126e-05	2.4568e-04

Table 2: Comparison of Numerical U and Exact U at time levels $t = 1.5$ and $t = 1.8$ respectively for $N = 50$, $\Delta t = 0.0001, \nu = 0.001$ and $c_0 = \frac{1}{2}$

x	Num. U $t = 1.5$	Exact U $t = 1.5$	Num. U $t = 1.8$	Exact U $t = 1.8$
0.0204	0.004	0.0038	0.0032	0.003
0.0408	0.0066	0.0064	0.0054	0.0052
0.0612	0.0073	0.0073	0.0062	0.0062
0.0816	0.0062	0.0065	0.0056	0.0058
0.102	0.0043	0.0046	0.0043	0.0046

Table 3: Assessment of L_2 and L_∞ errors at mentioned time level for $N = 201, \Delta t = 0.01, \nu = 0.001$ and $c_0 = \frac{1}{2}$

t	QBDQM [24]		RAMADAN [9]		PRESENT	
	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$
2	0.1370706	0.44538925	0.18354911	0.81852111	0.0708472	0.2592006
3	0.1168507	0.38428398	0.14414243	0.52348333	0.0619475	0.2233311
4	0.1019762	0.32583911	0.11441107	0.35635372	0.0557035	0.1896877
5	0.0920706	0.28166167	0.09478652	0.25497900	0.0514502	0.1624489
6	0.0849484	0.24842893	0.08141746	0.21348478	0.0482180	0.1450417
7	0.0794570	0.22254716	0.07189777	0.18800484	0.0455793	0.1295942
8	0.0750035	0.20195777	0.06483689	0.16826017	0.0433339	0.1183293
9	0.0712618	0.185151	0.05941149	0.15240749	0.0413755	0.1078538
10	0.0680382	0.17110335	0.05511514	0.13943121	0.0396394	0.1004443

Table 4: Order of Convergence at the time levels $t = 2$ and t = 2.5 respectively, where $\Delta t = 0.01$, $\nu = 0.001$ and $c_0 = \frac{1}{2}$

	Order of Convergence			
	$t = 2$	$t = 2$	$t = 2.5$	$t = 2.5$
	ROC L_2 ***U***	***ROC*** L_∞ ***U***	***ROC*** L_2 ***U***	***ROC*** L_∞ ***U***
$N_1 = 10$, $N_2 = 20$	*3.1244*	*2.7391*	*2.1921*	*2.9875*
$N_1 = 20$, $N_2 = 40$	*1.1125*	*1.1826*	*1.0022*	*1.1964*

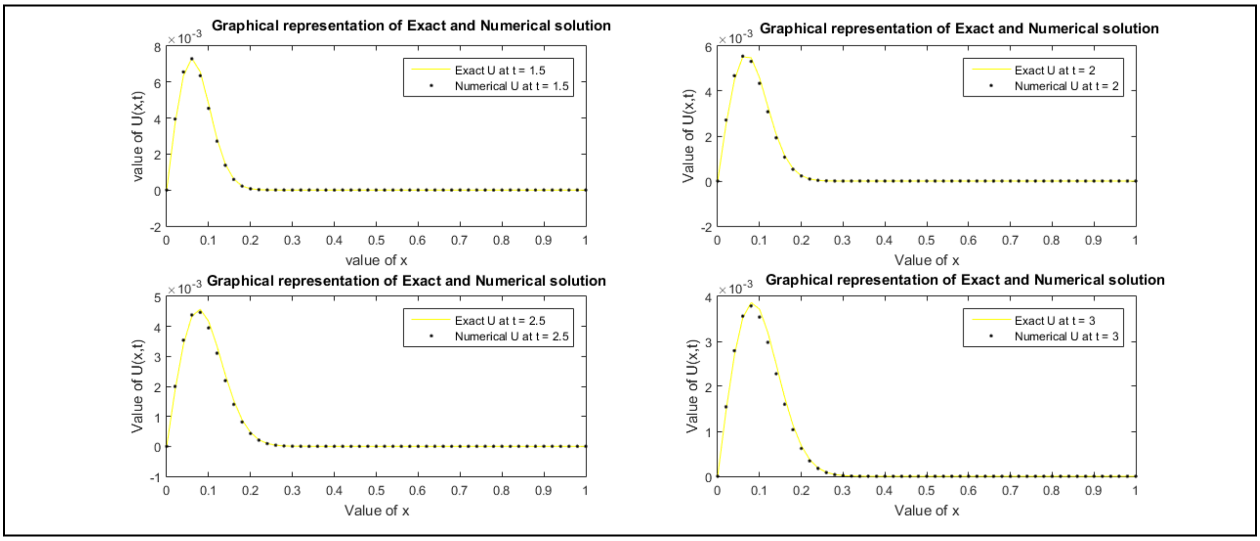

Figure 1: Graphical representation of Exact U(x, t) and Numerical U(x, t) at time levels t = 1.5, 2, 2.5 and 3 respectively for N = 51, $\Delta t = 0.01$, $\nu = 0.001$ and $c_0 = \frac{1}{2}$

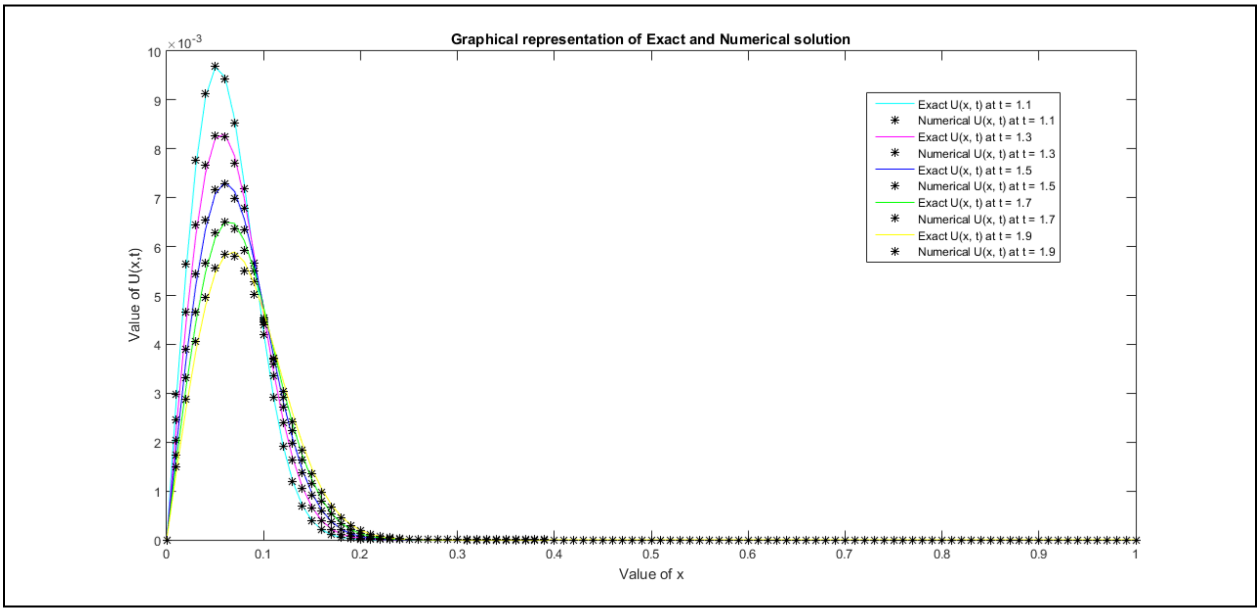

Figure 2: Graphical representation of Exact U(x, t) and Numerical U(x, t) at time levels t = 1.1, 1.2, 1.3, 1.4 and 1.5 respectively for N = 101, $\Delta t = 0.001$, $\nu = 0.001$ and $c_0 = \frac{1}{2}$

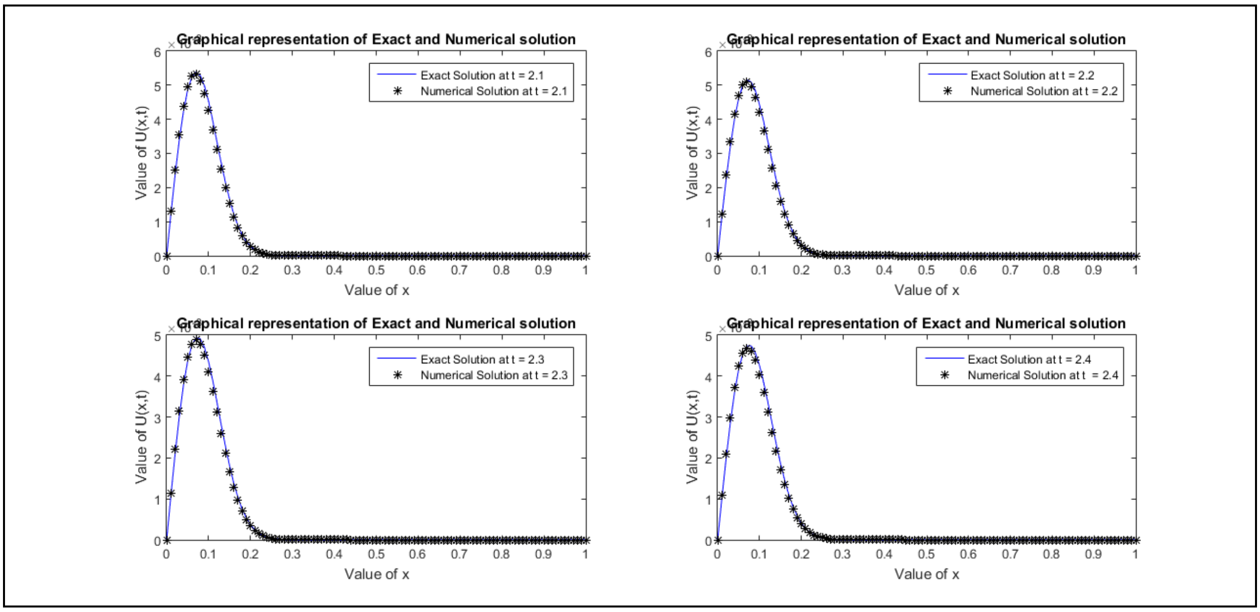

Figure 3: Pictorial depiction of Exact and Numerical Solutions at time levels t = 2.1, 2.2, 2.3 and 2.4 respectively for N = 101, $\Delta t = 0.001$, $\nu = 0.001$ and $c_0 = \frac{1}{2}$

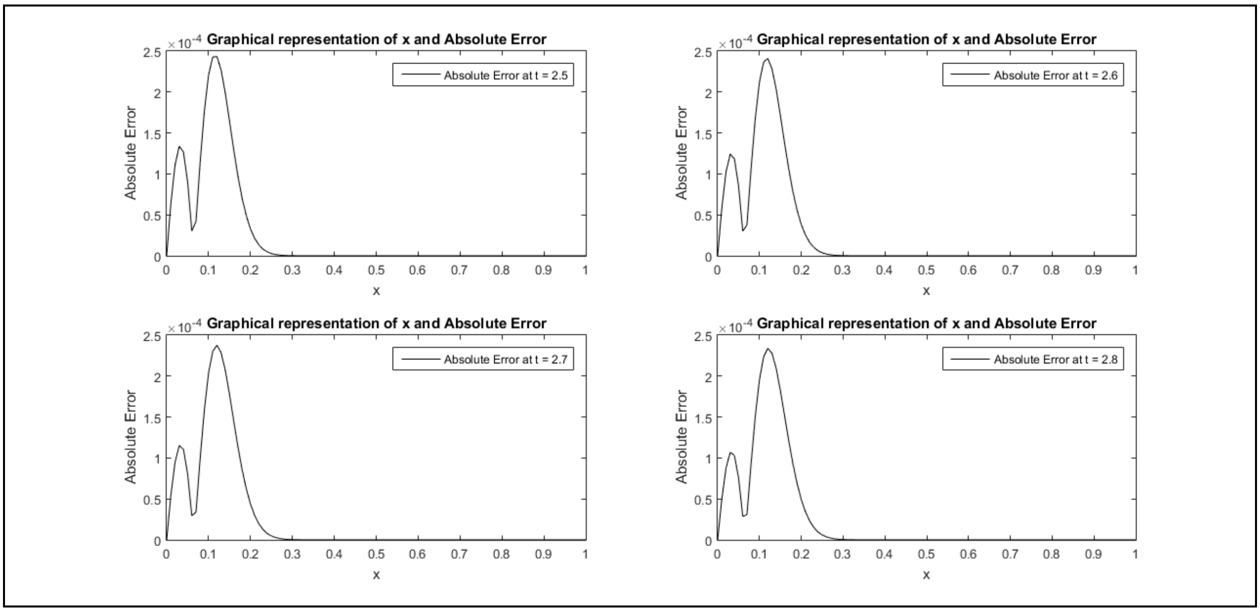

Figure 4: Graphical representation of the Absolute Error at time levels t = 2.5, 2.6, 2.7 and 2.8 respectively for N = 101, $\Delta t = 0.001$, $\nu = 0.001$ and $c_0 = \frac{1}{2}$

Table 5: Comparison of L_2 and L_∞ error norms with previous results

$\nu = 0.005$, $\Delta t = 0.001, h = 0.005$	$t = 2$	$t = 2$	$t = 10$	$t = 10$
	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$
Present	*2.2655e-01*	*5.8004e-01*	*1.3953e-01*	*2.2885e-01*
SFEM[24]	*0.02469*	*0.08456*	*0.12326*	*0.41604*
[10]	*0.25786*	*0.72264*	*0.18735*	*0.30006*
SBCM1[25]	*0.2289*	*0.58623*	*0.14042*	*0.23019*
SBCM2[25]	*0.23397*	*0.58424*	*0.13747*	*0.22626*
$\nu = 0.001$, $\Delta t = 0.01, h = 0.005$	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$
Present	*6.8170e-02*	*2.6094e-01*	*4.0653e-02*	*1.0259e-01*
SFEM[24]	*0.00549*	*0.0282*	*0.02743*	*0.13913*
QBDQM[24]	*0.13707*	*0.44538*	*0.06803*	*0.1711*

[9]	*0.18354*	*0.81852*	*0.05511*	*0.13943*
[10]	*0.06703*	*0.27967*	*0.0501*	*0.12129*
SBCM1[25]	*0.06843*	*0.26233*	*0.0408*	*0.10295*
SBCM2[25]	*0.0722*	*0.25975*	*0.03871*	*0.09882*
[12]	*0.06607*	*0.26186*	*0.0416*	*0.1047*
$\nu = 0.01, \Delta t = 0.01, h = 0.005$	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$
Present	*3.7937e-01*	*8.1690e-01*	*5.4503e-01*	*1.2812*
SFEM[24]	*0.09785*	*0.28062*	*0.48711*	*1.34692*
[10]	*0.52308*	*1.21698*	*0.64007*	*1.28124*
SBCM1[25]	*0.38489*	*0.82934*	*0.54826*	*1.28127*
SBCM2[25]	*0.39078*	*0.82734*	*0.54612*	*1.28127*
[12]	*0.37552*	*0.81766*	*0.19391*	*0.23074*
$\nu = 0.01, \Delta t = 0.01, h = 0.02$	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$
Present	*3.7973e-01*	*8.1250e-01*	*5.5092e-01*	*1.2812*
SFEM[24]	*0.09738*	*0.28025*	*0.48485*	*1.34798*
QBDQM[24]	*0.79558*	*1.37959*	*0.60429*	*1.47478*
[9]	*0.79042*	*1.70309*	*0.80025*	*1.80239*
SBCM1[25]	*0.38474*	*0.82611*	*0.55985*	*1.28127*
SBCM2[25]	*0.41321*	*0.81502*	*0.55095*	*1.28127*

Stability

Stability [21, 22, 26, 27, 28 ,29] of the proposed scheme is checked with the aid of matrix stability analysis method. It is found that the obtained eigen values are under the required range, which confirms that present regime is unconditionally stable.

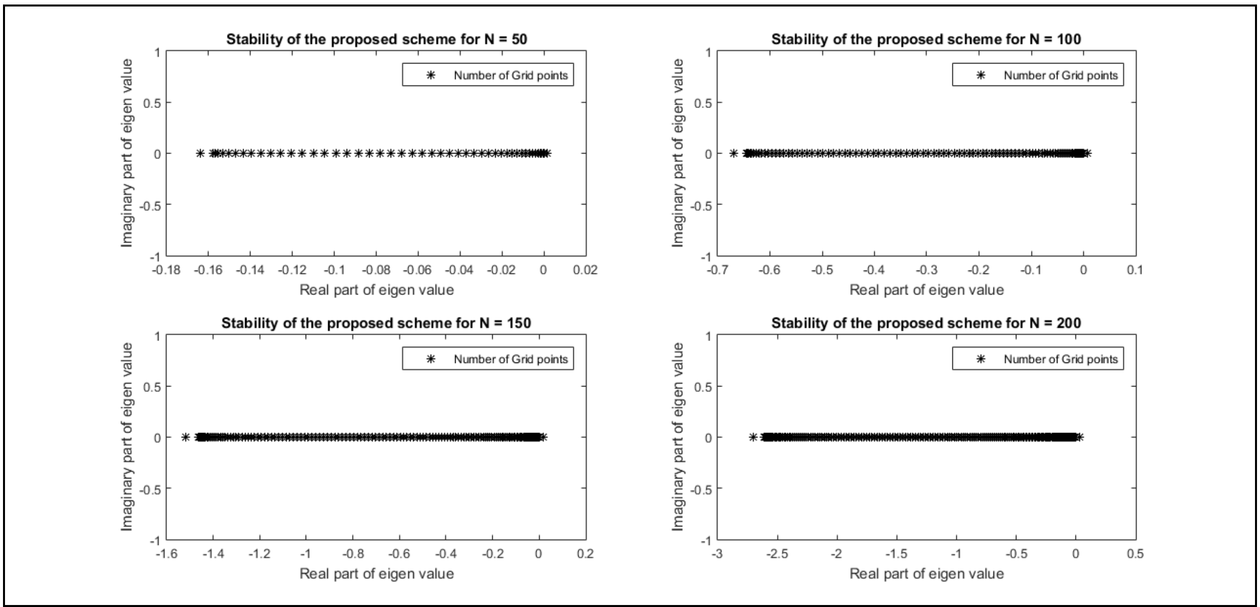

Figure 5: Stability of present regime for Number of grid points 50, 100, 150 and 200 respectively.

Conclusion

In present work, “modified cubic NUAH B-spline DQM” is implemented to solve Modified Burgers’ equation. In present paper, NUAH-B spline is considered as the basis function and weighting coefficients in DQM are obtained. After that, the reduced system of the equations is solve by using the SSP-RK43 scheme. This scheme has never been implemented in literature to solve the Modified Burgers’ equation. By checking effectiveness of proposed scheme with the aid of L_2 and L_∞ error norms, it is confirmed that the present results are acceptable. Order of convergence of the present method is also provided as well as the stability check is dome with the aid of matrix stability analysis method. It is confirmed that the proposed scheme is unconditionally stable. This numerical scheme is accurate, stable and easy to implement. It is obvious that it will help so many researchers in their future research work.

References

[1] H. Bateman (1915). "Some recent researches on the motion of fluids." *Monthly Weather Review*, *43*(4), 163-170, (1915).

[2] J. M. Burgers, "A mathematical model illustrating the theory of turbulence." In *Advances in applied mechanics* (Vol. 1, pp. 171-199). Elsevier, (1948).

[3] J. D. Cole, "On a quasi-linear parabolic equation occurring in aerodynamics." *Quarterly of applied mathematics*, *9*(3), 225-236, (1951).

[4] R. C. Mittal and P. Singhal, "Numerical solution of Burger's equation." *Communications in numerical methods in engineering*, *9*(5), 397-406, (1993).

[5] R. C. Mittal and P. Singhal, "Numerical solution of periodic Burgers equation." *Indian Journal of Pure and Applied Mathematics*, *27*, 689-700, (1996).

[6] T. Öziş, E. N. Aksan and A. Özdeş, "A finite element approach for solution of Burgers' equation." *Applied Mathematics and Computation*, *139*(2-3), 417-428, (2003).

[7] L. Gardner, G. A. Gardner and A. DOĞAN, "A Petrov-Galerkin finite element scheme for Burgers' equation,' (1997).

[8] S. E. L. C. U. K. Kutluay, A. Esen and I. Dag, "Numerical solutions of the Burgers' equation by the least-squares quadratic B-spline finite element method." *Journal of computational and Applied Mathematics*, *167*(1), 21-33, (2004).

[9] M. A. Ramadan, T. S. El-Danaf and F. E. Abd Alaal, "A numerical solution of the Burgers' equation using septic B-splines." *Chaos, Solitons & Fractals*, *26*(4), 1249-1258, (2005).

[10] M. A. Ramadan and T. S. El-Danaf, "Numerical treatment for the modified burgers equation." *Mathematics and Computers in Simulation*, *70*(2), 90-98, (2005).

[11] Y. Duan, R. Liu and Y. Jiang, "Lattice Boltzmann model for the modified Burgers' equation." *Applied Mathematics and Computation*, *202*(2), 489-497, (2008).

[12] T. Roshan and K. S. Bhamra, "Numerical solutions of the modified Burgers' equation by Petrov–Galerkin method." *Applied Mathematics and Computation*, *218*(7), 3673-3679, (2011).

[13] Z. Rong-Pei, Y. Xi-Jun and Z. Guo-Zhong, "Modified Burgers' equation by the local discontinuous Galerkin method." *Chinese Physics B*, *22*(3), 030210, (2013).

[14] A. G. Bratsos, "A fourth-order numerical scheme for solving the modified Burgers equation." *Computers & Mathematics with Applications*, *60*(5), 1393-1400, (2010).

[15] R. Bellman, B. G. Kashef and J. Casti, (1972). "Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations." *Journal of computational physics*, *10*(1), 40-52, (1972).

[16] R. Bellman, B. Kashef, E. S. Lee and R. Vasudevan, "Differential quadrature and splines." *Computers & Mathematics with Applications*, *1*(3-4), 371-376, (1975).

[17] A. Başhan, S. B. G. Karakoç and T. Geyikli, "B-spline differential quadrature method for the modified Burgers' equation." *Çankaya Üniversitesi Bilim ve Mühendislik Dergisi*, *12*(1), (2015).

[18] R. C. Mittal and R. Jiwari, "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, (2012).

[19] R. C. Mittal and S. Dahiya, "Numerical simulation on hyperbolic diffusion equations using modified cubic B-spline differential quadrature methods." *Computers & Mathematics with Applications*, *70*(5), 737-749, (2015).

[20] R. Jiwari, "Lagrange interpolation and modified cubic B-spline differential

quadrature methods for solving hyperbolic partial differential equations with Dirichlet and Neumann boundary conditions." *Computer Physics Communications*, *193*, 55-65, (2015).

[21] 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*, *9*(45), (2016).

[22] 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*, *57*(2), 1087-1098, (2018).

[23] R. C. Mittal and S. Dahiya, "A comparative study of modified cubic B-spline differential quadrature methods for a class of nonlinear viscous wave equations." *Engineering Computations*, (2018).

[24] S. B. G. Karakoç, A. Başhan and T. Geyikli, "Two different methods for numerical solution of the modified Burgers' equation." *The Scientific World Journal*, (2014).

[25] D. Irk, "Sextic B-spline collocation method for the modified Burgers' equation." *Kybernetes*, (2009).

[26] 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).

[27] R. C. Mittal, R. Jiwari and K. K. Sharma, "A numerical scheme based on differential quadrature method to solve time dependent Burgers' equation.", *Engineering Computations*, (2013).

[28] A. Korkmaz and İ Dağ, "Cubic B-spline differential quadrature methods and stability for Burgers' equation.", *Engineering Computations*, , vol. 30, no. 3, pp. 320-344, (2013),

[29] M. K. Jain, "Numerical solution of differential equations", 2nd edition., Wiley, New York, NY, **(1983)**.