HOSTED BY

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/bjbas

Full Length Article

# Extended modified cubic B-spline algorithm for nonlinear Burgers' equation

*Mohammad Tamsir* [a,*], *Neeraj Dhiman* [a,b], *Vineet K. Srivastava* [c,d]

[a] *Department of Mathematics, Graphic Era University, Dehradun 248001, India*
[b] *Department of Mathematics, Graphic Era Hill University, Dehradun 248001, India*
[c] *Flight Dynamics Group, ISRO Telemetry Tracking and Command Network, Bangalore 560058, India*
[d] *Department of Applied Mathematics, Indian Institute of Technology (Indian School of Mines) Dhanbad, Dhanbad 826004, India*

ARTICLE INFO

*Article history:*
Received 14 March 2016
Received in revised form 13 July 2016
Accepted 3 September 2016
Available online 19 September 2016

*Keywords:*
Extended modified cubic B-spline functions
DQM
Burgers' equation
SSP-RK54
Stability analysis

ABSTRACT

In this paper, an extended modified cubic B-Spline differential quadrature method is proposed to approximate the solution of the nonlinear Burgers' equation. The proposed method is used in space and a five-stage and four order strong stability-preserving time-stepping Runge–Kutta (SSP-RK54) method is used in time. The accuracy and efficiency of the method is illustrated by considering four numerical problems. The numerical results of the method are compared with some existing methods and it was found that the proposed numerical method produces acceptable results and even more accurate results in comparison with some existing methods. The stability analysis of the scheme is also carried out and was found to be unconditionally stable.



## 1. Introduction

Consider one dimensional nonlinear Burgers' equation

$$\frac{\partial u}{\partial t}+\alpha u\frac{\partial u}{\partial x}-\upsilon\frac{\partial^2 u}{\partial x^2}=0;\quad a_1\le x\le a_2,\quad t\ge 0, \tag{1.1}$$

with initial conditions:

$$u(x,0)=\phi(x);\quad a_1\le x\le a_2, \tag{1.2}$$

and Dirichlet boundary conditions:

$$u(a_1,t)=0=u(a_2,t);\, t\ge 0, \tag{1.3}$$

* *Corresponding author.* Department of Mathematics, Graphic Era University, Dehradun 248001, India.
*E-mail address:* tamsiriitm@gmail.com (M. Tamsir).
http://dx.doi.org/10.1016/j.bjbas.2016.09.001

and two dimensional nonlinear coupled Burgers' equations

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=\upsilon\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right), \quad (1.4)$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=\upsilon\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right), \quad (1.5)$$

with initial conditions:

$$u(x,y,0)=\varphi_1(x,y) \text{ and } v(x,y,0)=\varphi_2(x,y), \quad (x,y)\in\Omega, \quad (1.6)$$

and Dirichlet boundary conditions:

$$u(x,y,t)=\psi_1(x,y,t) \text{ and } v(x,y,t)=\psi_2(x,y,t), \quad (x,y)\in\partial\Omega, t>0, \quad (1.7)$$

where $\Omega=\{(x,y): a_1\le x\le a_2, \quad b_1\le x\le b_2\}$ is the computational square domain and $\partial\Omega$ is its boundary, $u(x,t)$ is the velocity component in one dimension, $u(x,y,t)$ and $v(x,y,t)$ are the velocity components in two dimension; $\phi$, $\varphi_1$, $\varphi_2$, $\psi_1$ and $\psi_2$ are known functions; $\frac{\partial u}{\partial t}$ is unsteady term, $u\frac{\partial u}{\partial x}$ is the nonlinear convection term, $\upsilon\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)$ is the diffusion term and $\upsilon>0$ is the coefficient of viscosity and $\alpha$ is the some positive constant.

The Burgers' equation is an easiest model for explaining various physical flows problems, such as hydrodynamic turbulence, sound and shock wave theory, vorticity transportation, dispersion in porous media, modeling of turbulent fluid, etc, Burgers (1939), Cole (1951), Esipov (1995). First of all, Bateman (1915) introduced this type of equations and later a steady-state solution was proposed by Burgers (1939).

The Burgers' equation has been solved by various analytical and numerical schemes such as Hofe Cole transformation (Cole, 1951; Fletcher, 1983),finite element method (Aksan, 2006; Cecchi et al., 1996; Dogan and Galerkin, 2004; Ozis et al., 2003), finite difference method (Hassanien et al., 2005), explicit finite difference method (Kutluay et al., 1999), implicit finite difference method (Kadalbajoo et al., 2005), compact finite difference method (Liao, 2008), implicit logarithmic finite difference method (Srivastava et al., 2013), least-squares quadratic B-spline finite element method (Kutluay et al., 2004), quadratic B-spline finite elements (Ozis et al., 2005), B-Spline collocation method (Dag et al., 2005), quartic B-spline collocation method (Saka and Dag, 2007), reproducing kernel function method (Xie et al., 2008), cubic B-spline quasi-interpolation (Jiang and Wang, 2010), sinc differential quadrature method (Korkmaz and Dag, 2011), Fourier expansion-based differential quadrature method (Shu and Chew, 1997), quartic B-spline differential quadrature method (Korkmaz et al., 2011), modified cubic B-splines collocation method (Mittal and Jain, 2012), cubic B-spline differential quadrature methods (Korkmaz and Dag, 2013), and modified cubic B-spline differential quadrature methods (Arora and Singh, 2013; Shukla et al., 2014).

Doha et al. (2014) presented a numerical solution of the nonlinear coupled viscous Burgers' equation based on spectral methods. They employed a Jacobi–Gauss–Lobatto collocation scheme in combination with the implicit Runge–Kutta–Nyström scheme. Bhrawy et al. (2015) proposed new space-time spectral algorithm based on spectral shifted Legendre collocation method in combination with the shifted Legendre operational matrix of fractional derivatives to approximate the solution for the space-time fractional Burgers' equation. Bhrawy and Zaky (2016) proposed a new approach based on the shifted Chebyshev polynomials to approximate the solution of functional variable-order fractional differential equations. Esen and Tasbozan (2016) used collocation method based on cubic B-spline basis functions for the time fractional Burgers' equation.

Bellman et al. (1972) were the first to introduce an efficient technique named "differential quadrature method (DQM)" for the solution of PDEs. It was further improved by Quan and Chang (1989) to approximate the weighting coefficients. In DQM, several kinds of test functions have been used to compute the weighting coefficients viz. B-spline function, cubic B-spline functions, sinc function, Lagrange interpolation polynomials, Legendre polynomials, quartic B-spline functions, modified cubic B-spline functions, etc. B-splines are a set of certain functions that can be used to build piece-wise polynomial by computing the suitable linear combination. These basis functions have more influence in comparison to other basis functions due to its smoothness and capability to handle local phenomena. Recently, Dag et al. (2013) proposed an extended cubic B-spline algorithm for numerical solution of a modified regularized long wave equation.

The main objective of this study is to present a new method, namely, an extended modified cubic-B-spline differential quadrature method (EMCB-DQM) for the numerical computation of the Burgers' equation. In this method, the extended modified cubic-B-spline basis functions are used in DQM to determine the weighting coefficients which transform the Burgers' equation into a system of first order ordinary differential equations. The resulting system of equations is solved by employing a five-stage and four order strong stability-preserving time-stepping Runge–Kutta method. The efficacy and adaptability of the method is confirmed by taking a four test problem in one and two dimensions. The rest of the paper is prepared as: in Section 2, the EMCB-DQM is introduced; in Section 3, implementation procedure to the Burgers' equation is illustrated; in Section 4, stability analysis is discussed; in Section 5, four test problems are considered in order to establish the applicability and accuracy of the proposed method, while Section 6 concludes our study.

## 2. Extended modified cubic B-spline differential quadrature method

For one dimensional Burgers' equation, let us assume that $N$ grid points/knots $a_1=x_1<x_2,\ldots..<x_N=a_2$ are uniformly distributed with step size $\Delta x=x_{i+1}-x_i$ along $x$ direction. The rth order spatial partial derivatives of the unknown $u(x,t)$ with respect to $x$, approximated at $x_i$, $i=1,2,\ldots,N$, are defined as

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

For two dimensional case, it is assumed that $N$ and $M$ grid points: $a_1 = x_1 < x_2, \ldots < x_N = a_2$ and $b_1 = y_1 < y_2, \ldots < y_M = b_2$ are uniformly distributed with step size $\Delta x = \frac{a_2 - a_1}{N-1}$ and $\Delta y = \frac{b_2 - b_1}{M-1}$ in $x$ and $y$ directions, respectively.

The rth order spatial partial derivatives of $u(x, y, t)$ with respect to $x$ (keeping $y_j$ as fixed) and with respect to $y$ (keeping $x_i$ as fixed), approximated at $x_i$ and $y_j$, respectively, are defined as:

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

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

Similarly, the rth order spatial partial derivatives of $v(x, y, t)$ with respect to $x$ and with respect to $y$ are approximated as:

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

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

where $a_{ij}^{(r)}$ and $b_{ij}^{(r)}$, $r = 1,2$ are the weighting coefficients of the rth order spatial partial derivatives with respect to $x$ and $y$.

### 2.1. *Extended cubic B-spline basis functions*

The extended cubic B-spline basis functions at the grid points are defined as (Dag et al., 2013)

$$E_m(x) = \frac{1}{h^3}\begin{cases} 4h(1-\lambda)(x - x_{m-2})^3 + 3\lambda(x - x_{m-2})^4, & x \in [x_{m-2}, x_{m-1}) \\ h^4(4-\lambda) + 12h^3(x - x_{m-1}) + 6h^2(2+\lambda)(x - x_{m-1})^2 \\ \quad - 12h(x - x_{m-1})^3 - 3\lambda(x - x_{m-1})^4, & x \in [x_{m-1}, x_m) \\ h^4(4-\lambda) - 12h^3(x - x_{m+1}) + 6h^2(2+\lambda)(x - x_{m+1})^2 \\ \quad + 12h(x - x_{m+1})^3 - 3\lambda(x - x_{m+1})^4, & x \in [x_m, x_{m+1}) \\ 4h(\lambda-1)(x - x_{m+2})^3 + 3\lambda(x - x_{m+2})^4, & x \in [x_{m+1}, x_{m+2}), \\ 0, & \text{otherwise}, \end{cases} \tag{2.6}$$

In Eq. (2.6), the free parameter $\lambda$ is used to obtain different forms of extended cubic B-spline functions. For $\lambda = 0$, the extended cubic B-spline functions reduce to cubic B-spline functions. $\{E_0, E_1, \ldots, E_N, E_{N+1}\}$ is chosen in such a way that it forms a basis over the domain $a_1 \le x \le a_2$. The values of extended cubic B-splines and its derivatives at the nodal points are depicted in Table 1.

**Table 1 – Coefficients of the extended cubic B-spline $E_m$ and its derivatives at the node $x_m$.**

| | $x_{m-2}$ | $x_{m-1}$ | $x_m$ | $x_{m+1}$ | $x_{m+2}$ |
|---|---|---|---|---|---|
| $24E_m(x)$ | 0 | $4-\lambda$ | $16+2\lambda$ | $4-\lambda$ | 0 |
| $2hE'_m(x)$ | 0 | 1 | 0 | -1 | 0 |
| $2h^2E''_m(x)$ | 0 | $2+\lambda$ | $-4-2\lambda$ | $2+\lambda$ | 0 |

It was modified in such a way that it reduces the matrix size and complexity, thus, requires less number of grid points. The extended cubic B-spline basis functions are modified as

$$\left.\begin{aligned} \omega_1(x) &= E_1(x) + 2E_0(x) \\ \omega_2(x) &= E_2(x) - E_0(x) \\ \omega_m(x) &= E_m(x) \ \text{for}\ m = 3, \ldots, N-2 \\ \omega_{N-1}(x) &= E_{N-1}(x) - E_{N+1}(x) \\ \omega_N(x) &= E_N(x) + 2E_{N+1}(x) \end{aligned}\right\}, \tag{2.7}$$

where $\{\omega_1, \omega_2, \ldots, \omega_N\}$ forms a basis in the region $a_1 \le x \le a_2$.

### 2.2. *Computation of the weighting coefficients*

Taking $r = 1$ in Eq. (2.1) and substituting the values of $\omega_m(x)$, $m = 1, 2, \ldots, N$, we get a system of linear equations:

$$\omega'_m(x_i) = \sum_{j=1}^{N} a_{ij}^{(1)} \omega_m(x_j), \text{for } i, m = 1, 2, \ldots, N, \tag{2.8}$$

with the help of Eq. (2.7) and Table 1, Eq. (2.8) reduces into a tridiagonal system of equations:

$$A\vec{a}^{(1)}[i] = \vec{R}[i],\ \text{for}\ i = 1, 2, \ldots, M, \tag{2.9}$$

where $A$ is the coefficient matrix of order $N$ given by:

$$A = \begin{bmatrix} 1 & \frac{4-\lambda}{24} & & & & & \\ 0 & \frac{16+2\lambda}{24} & \frac{4-\lambda}{24} & & & & \\ & \frac{4-\lambda}{24} & \frac{16+2\lambda}{24} & \frac{4-\lambda}{24} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \frac{4-\lambda}{24} & \frac{16+2\lambda}{24} & \frac{4-\lambda}{24} & \\ & & & & \frac{4-\lambda}{24} & \frac{16+2\lambda}{24} & 0 \\ & & & & & \frac{4-\lambda}{24} & 1 \end{bmatrix}$$

$\vec{a}^{(1)}[i]=\left[a_{i1}^{(1)},a_{i2}^{(1)},\dots,a_{iN}^{(1)}\right]^T$ is the weighting coefficient vector corresponding to the knot point $x_i$, and the coefficient vector $\vec{R}[i]=[\omega'_{1,i},\omega'_{2,i},\dots\dots,\omega'_{N-1,i},\omega'_{N,i}]^T$ corresponding to the knot point $x_i$, $i=1,2,\dots,N$ are evaluated as:

$$\vec{R}[1]=\begin{bmatrix}-\frac{1}{h}\\ \frac{1}{2h}\\ 0\\ 0\\ \vdots\\ 0\\ 0\end{bmatrix},\ \vec{R}[2]=\begin{bmatrix}-\frac{1}{2h}\\ 0\\ \frac{1}{2h}\\ 0\\ \vdots\\ 0\\ 0\end{bmatrix},\ \vec{R}[3]=\begin{bmatrix}0\\ -\frac{1}{2h}\\ 0\\ \frac{1}{2h}\\ 0\\ \vdots\\ 0\end{bmatrix},\dots,\ \vec{R}[N-1]=\begin{bmatrix}0\\ 0\\ \vdots\\ 0\\ -\frac{1}{2h}\\ 0\\ \frac{1}{2h}\end{bmatrix},$$

$$\vec{R}[N]=\begin{bmatrix}0\\ 0\\ \vdots\\ 0\\ 0\\ -\frac{1}{h}\\ \frac{1}{h}\end{bmatrix}.$$

We note that the coefficient matrix $A$ is invertible. The tridiagonal system of equations is solved for each knot point $x_i$ $(i=1,2,\dots,N)$ using the Thomas algorithm, which gives the weighting coefficients $a_{i1}^{(1)},a_{i2}^{(1)},\dots,a_{iN-1}^{(1)},a_{iN}^{(1)}$ $(i=1,2,\dots,N)$ for the first order partial derivative.

In a similar way, the weighting coefficients $a_{ij}^{(2)},1\le i,j\le N$ for the second order partial derivative, are determined. Weighting coefficients $a_{ij}^{(2)},1\le i,j\le N$, can also be computed by using the formula (Shu, 2000)

$$\begin{cases}a_{ij}^{(r)}=r\left(a_{ij}^{(1)}a_{ii}^{(r-1)}-\dfrac{a_{ij}^{(r-1)}}{x_i-x_j}\right),\ for\ j\ne i \text{ and } i=1,2,3,\dots,N;\\ \quad r=2,3,\dots,N-1,\\ a_{ii}^{(r)}=-\displaystyle\sum_{j=1,j\ne i}^{N}a_{ij}^{(r)},\ for\ i=j,\end{cases}\tag{2.10}$$

where $a_{ij}^{(r-1)}$and $a_{ij}^{(r)}$ are the weighting coefficients of the $(r-1)^{th}$ and $r^{th}$ order partial derivatives with respect to $x$.

In the same manner, the weighting coefficients $b_{ij}^{(1)}$ of the first order partial derivatives with respect to $y$ is obtained. Weighting coefficients $b_{ij}^{(2)},1\le i,j\le N$ for the second derivatives can also be computed from the formula:

$$\begin{cases}b_{ij}^{(r)}=r\left(b_{ij}^{(1)}b_{ii}^{(r-1)}-\dfrac{b_{ij}^{(r-1)}}{x_i-x_j}\right),\ for\ j\ne i \text{ and } i=1,2,3,\dots,N;\\ \quad r=2,3,\dots,N-1,\\ b_{ii}^{(r)}=-\displaystyle\sum_{j=1,j\ne i}^{N}b_{ij}^{(r)},\ for\ i=j,\end{cases}\tag{2.11}$$

where $b_{ij}^{(r-1)}$ and $b_{ij}^{(r)}$ are the weighting coefficients of the $(r-1)^{th}$ and $r^{th}$ order partial derivatives with respect to $y$.

## 3. Implementation of the method to the Burgers' equation

After discretizing the spatial derivatives of the Burgers' equation (1.1) by EMCB-DQM, we get the following system of nonlinear ordinary differential equations

$$\frac{du(x_i,t)}{dt}=-\alpha u(x_i)\sum_{j=1}^{N}a_{ij}^{(1)}u(x_j)-\upsilon\sum_{j=1}^{N}a_{ij}^{(2)}u(x_j),\ a\le x\le b,\ t>0 \text{ and } i=1,2,\dots,N,\tag{3.1}$$

(3.1) Eq. (3.1) can be written as

$$\frac{du(x_i,t)}{dt}=L(u(x_i,t)),\ i=1,2,\dots,N,\tag{3.2}$$

with the initial and boundary conditions (1.2) and (1.3).

On substituting the approximated values of the spatial derivatives computed by EMCB-DQM, Eq. (1.4) can be written as

$$\begin{aligned}\frac{du(x_i,y_j,t)}{dt}&=-u(x_i,y_j)\sum_{k=1}^{N}a_{ik}^{(1)}u(x_k,y_j)-v(x_i,y_j)\sum_{k=1}^{M}b_{jk}^{(1)}u(x_i,y_k)\\&\quad+\upsilon\left[\sum_{k=1}^{N}a_{ik}^{(2)}u(x_k,y_j)+\sum_{k=1}^{M}b_{jk}^{(2)}u(x_i,y_k)\right],\\&\quad(x_i,y_j)\in\Omega,\ t>0,\ i=1,2,\dots,N,\quad j=1,2,\dots,M.\end{aligned}\tag{3.3}$$

Similarly, Eq. (1.5) can be written as

$$\begin{aligned}\frac{du(x_i,y_j,t)}{dt}&=-u(x_i,y_j)\sum_{k=1}^{N}a_{ik}^{(1)}v(x_k,y_j)-v(x_i,y_j)\sum_{k=1}^{M}b_{jk}^{(1)}v(x_i,y_k)\\&\quad+\upsilon\left[\sum_{k=1}^{N}a_{ik}^{(2)}v(x_k,y_j)+\sum_{k=1}^{M}b_{jk}^{(2)}v(x_i,y_k)\right],\\&\quad(x_i,y_j)\in\Omega,\ t>0,\ i=1,2,\dots,N,\quad j=1,2,\dots,M.\end{aligned}\tag{3.4}$$

Eq. (3.3) and Eq. (3.4) are reduced into the following system of nonlinear first order ordinary differential equations

$$\frac{du(x_i,y_j,t)}{dt}=F_1(u(x_i,y_j,t)),\quad i=1,2,\dots,N \text{ and } j=1,2,\dots,M,\tag{3.5}$$

and

$$\frac{dv(x_i,y_j,t)}{dt}=F_2(u(x_i,y_j,t)),\quad i=1,2,\dots,N \text{ and } j=1,2,\dots,M,\tag{3.6}$$

where $L$, $F_1$ and $F_2$ denote spatial nonlinear differential operators. SSP-RK54 scheme (Gottlieb et al., 2009) is used to solve Eq. (3.2) together with initial conditions boundary conditions. Similarly, Eqs. (3.5) and (3.6) can be solved together with appropriate initial and boundary conditions.

## 4. Stability analysis

After discretization via DQM and linearization of the nonlinear term $uu_x$, $uu_x+vu_y$ and $uv_x+vv_y$ by assuming $u$ and $v$

locally constant (Saka et al., 2009), Eq. (3.1) is reduced into a set of ordinary differential equations in time:

$$\frac{dU}{dt} = PU + E, \tag{4.1}$$

Eq. (3.3) and (3.4) are reduced into set of ordinary differential equations in time:

$$\frac{d\{R\}}{dt} = T\{R\} + \{K\}, \tag{4.2}$$

where, $T = \begin{bmatrix} A & O \\ O & B \end{bmatrix}$,

and

(i) $U$ is an unknown vector of the functional values at the interior grid points:

$$U = (u_2, u_3, \ldots, u_{N-1})$$

(ii) $E$ is a vector containing nonhomogeneous part and boundary conditions.
(iii) $P = -\alpha U_{ij} P_1 + \upsilon P_2$
(iv) $O$'s are null matrices.
(v) $\{K\} = (F, G)^T$ is a vector containing nonhomogeneous part and boundary conditions.
(vi) $\{R\} = (V, W)^T$ is an unknown vector of the functional values at the interior grid points:

$$V = (u_{22}, u_{23}, \ldots, u_{2(M-1)}, u_{32}, u_{33}, \ldots u_{3(M-1)}, \ldots u_{(M-1)2}, u_{(M-1)3} \ldots, u_{(M-1)(M-1)}),$$
$$W = (v_{22}, v_{23}, \ldots, v_{2(M-1)}, v_{32}, v_{33}, \ldots v_{3(M-1)}, \ldots v_{(M-1)2}, v_{(M-1)3}, \ldots, v_{(M-1)(N-1)}).$$

(vii) $A = -U_{ij}A_1 - V_{ij}B_1 + \upsilon A_2 + \upsilon B_2$ and $B = -U_{ij}A_1' - V_{ij}B_1' + \upsilon A_2' + \upsilon B_2'$,

where $P_r$ are matrices of the weighting coefficients $a_{ij}^{(r)}$, $(r = 1,2)$ respectively and given by

$$P_r = \begin{bmatrix} a_{22}^{(r)} & a_{23}^{(r)} & \cdots & a_{2(N-1)}^{(r)} \\ a_{32}^{(r)} & a_{33}^{(r)} & \cdots & a_{3,N-1}^{(r)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{(N-1)2}^{(r)} & a_{(N-1)3}^{(r)} & \cdots & a_{(N-1)(N-1)}^{(r)} \end{bmatrix}_{(N-2)\times(N-2)}.$$

$A_r$ and $B_r$ are square block diagonal matrices (each of order $(N-2)(M-2)$) of the weighting coefficients $a_{ij}^{(r)}, b_{ij}^{(r)}$ $(r = 1,2)$, respectively, given by:

$$A_r = \begin{bmatrix} a_{22}^{(r)}I & a_{23}^{(r)}I & \cdots & a_{2(N-1)}^{(r)}I \\ a_{32}^{(r)}I & a_{33}^{(r)}I & \cdots & a_{3,N-1}^{(r)}I \\ \vdots & \vdots & \ddots & \vdots \\ a_{(N-1)2}^{(r)}I & a_{(N-1)3}^{(r)}I & \cdots & a_{(N-1)(N-1)}^{(r)}I \end{bmatrix}$$

$$B_r = \begin{bmatrix} M_r & O & \cdots & O \\ O & M_r & \cdots & O \\ \vdots & \vdots & \ddots & \vdots \\ O & O & \cdots & M_r \end{bmatrix}; \quad \text{where}$$

$$M_r = \begin{bmatrix} b_{22}^{(r)} & b_{23}^{(r)} & \cdots & b_{2(M-1)}^{(r)} \\ b_{32}^{(r)} & b_{33}^{(r)} & \cdots & b_{3(M-1)}^{(r)} \\ \vdots & \vdots & \ddots & \vdots \\ b_{(M-1)2}^{(r)} & b_{(M-1)3}^{(r)} & \cdots & b_{(M-1)(M-1)}^{(r)} \end{bmatrix}$$

$I$ and $O$ are the matrices of order $(N-2)$ and $(M-2)$.

Similarly, $A_r'$ and $B_r'$ are square block diagonal matrices (each of order $(N-2)(M-2)$) of the weighting coefficients $a_{ij}^{(r)}$ and $b_{ij}^{(r)}$ $(r = 1,2)$, respectively.

Stability of the proposed scheme for the solution of nonlinear coupled viscous Burgers' equation directly depends upon the stability of the system of ordinary differential Eq. (4.1) in one dimension and on Eq. (4.2) in two dimensions. Stability of (4.1) and (4.2) depends on the Eigen values of the coefficient matrices $P$ and $T$. The system (4.1) and (4.2) will be stable if the real part of each Eigen value of $P$ and $T$ are either negative or zero.

It is clear from Fig. 1 that the real part of the Eigen values of the matrices $P_1$ and $P_2$ are either negative or zero for different grid sizes. Since the Eigen values of the matrix $P$ depends upon the Eigen values of the matrices $P_1$ and $P_2$, therefore, the

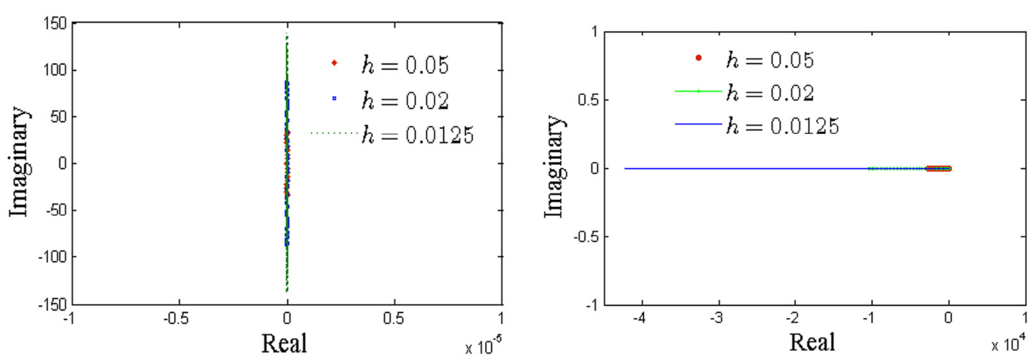


**Fig. 1 – Eigen values of $P_1$ (left) and $P_2$ (right) for different grid sizes.**

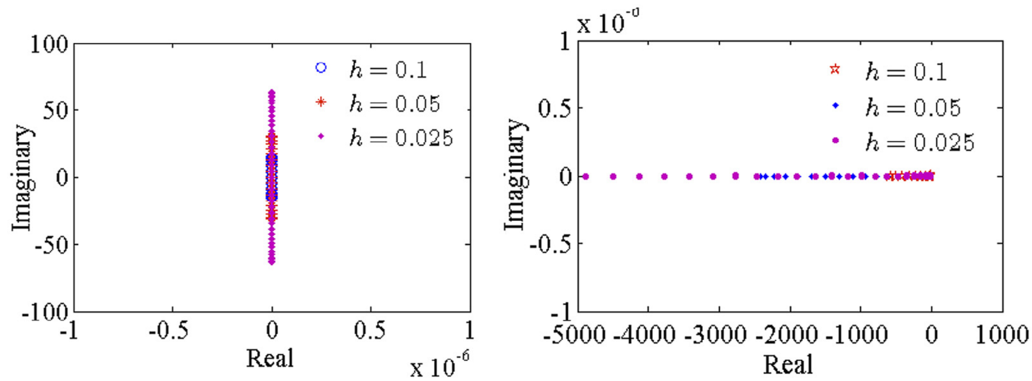


**Fig. 2 – Eigen values of $A_1$ (left) and $A_2$ (right) for different grid sizes.**

real part of all Eigen values of the matrices $P$ are either negative or zero. Again, it is clear from Figs. 2 and 3 that the real part of the Eigen values of the matrices $A_1$,$A_2$,$B_1$ and $B_2$ are either negative or zero for different grid sizes. Since the Eigen values matrices $A$ and $B$ depend upon the Eigen values matrices $A_1$,$A_2$,$B_1$ and $B_2$, therefore, the real part of all Eigen values of the matrices $A$ and $B$ are either negative or zero and hence the Eigen values of matrix $T$ are either negative or zero. This shows that the proposed method is unconditionally stable.

## 5. Numerical results and discussion

In this section, four numerical problems of the Burgers' equation are considered to show the accuracy and efficiency of the proposed method. The error norms $L_2$ and $L_\infty$ are calculated by using the following definitions

$$\left.\begin{aligned} L_2 &:= \|u_{exact} - u_{computed}\|_2 = \sqrt{h\sum_{j=1}^{n}\left|u_j^{exact} - u_j^{computed}\right|^2} \\ L_\infty &:= \|u_{exact} - u_{computed}\|_\infty = \max_j\left|u_j^{exact} - u_j^{computed}\right| \end{aligned}\right\}, \qquad (5.1)$$

where $u_{exact}$ and $u_{computed}$ denote the exact and computed solutions at the node $j$, respectively.

Example 1. In this problem, the Burgers' equation (1.1) with $\alpha = 1$ is solved over the region [0, 1.2], and the initial and boundary conditions are taken as (Aksan, 2006; Arora and Singh, 2013)

$$u(x,1) = \frac{x}{1+\exp\left(\frac{1}{4\upsilon}\left(x^2 - \frac{1}{4}\right)\right)}, \quad \text{with } u(0,t) = 0,\ u(1.2,t) = 0,$$

for $t > 1$.

The initial condition is taken at $t = 1$ and the exact solution is given by:

$$u(x,1) = \frac{\frac{x}{t}}{1+\left(\frac{t}{t_0}\right)^{1/2}\exp\left(\frac{x^2}{4\upsilon t}\right)}, \quad \text{where } t_0 = \exp\left(\frac{1}{8\upsilon}\right), \text{for } t \geq 1. \qquad (5.2)$$

For Ex. 1, the computation is performed for different values of $t$ ranging from 1.7 to 2.5 with $\Delta t = 0.01$ $h = 0.01$ in order to find the best value of $\lambda$ over the interval [–2, 2]. The corresponding maximum absolute error at $t = 2.5$ is shown in Fig. 4. It is found that the least maximum absolute error is obtained when $\lambda = -0.012$. Comparison of the proposed EMCB-DQM scheme, in terms of $L_2$ and $L_\infty$ errors, are shown in Tables 2 and 3 at different time levels with those obtained by Arora and Singh (2013), Chung et al. (2010), Dag et al. (2005), Korkmaz et al. (2011),

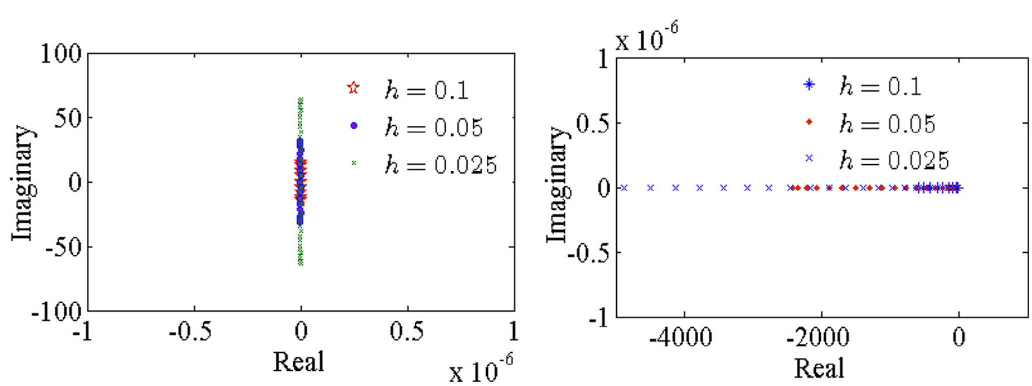


**Fig. 3 – Eigen values of $B_1$ (left) and $B_2$ (right) for different grid sizes.**

**Table 2 – Comparison of $L_2$ and $L_\infty$ error in EMCB-DQM solution for $\upsilon = 0.005$, $\lambda = -0.012$, $h = 0.01$ with the errors obtained in earlier schemes at different time levels.**

| Methods | N | Δt | t = 1.7 | | t = 2.4 | | t = 3.1 | |
|---|---|---|---|---|---|---|---|---|
| | | | $L_{2\times10^3}$ | $L_{\infty\times10^3}$ | $L_{2\times10^3}$ | $L_{\infty\times10^3}$ | $L_{2\times10^3}$ | $L_{\infty\times10^3}$ |
| Present | 121 | 0.01 | 0.00101 | 0.00484 | 0.00062 | 0.00199 | 0.00070 | 0.00354 |
| MCB-DQM (Arora and Singh, 2013) | 121 | 0.01 | 0.00191 | 0.00777 | 0.00086 | 0.00308 | 0.00065 | 0.00331 |
| QRTDQ (Korkmaz et al., 2011) | 101 | 0.001 | 0.109 | 0.434 | 0.100 | 0.339 | 0.091 | 0.266 |
| BS.FEM (Chung et al., 2010) | 50 | 0.1 | 0.857 | 2.576 | 0.423 | 1.242 | 0.230 | 0.680 |
| C.S.C. (Salas, 2010) | 50 | 0.01 | 0.857 | 2.576 | 0.423 | 1.242 | 0.235 | 0.688 |
| Galerkin (Zhang et al., 2010) | 200 | 0.01 | 0.857 | 2.576 | 0.423 | 1.242 | 0.235 | 0.688 |
| | | | | | t = 2.5 | | | |
| QBCM (Dag et al., 2005) | 200 | 0.01 | 0.0721 | 0.31153 | 0.0510 | 0.18902 | | |
| CBCM (Dag et al., 2005) | 200 | 0.01 | 2.4664 | 27.577 | 2.1118 | 25.1517 | | |
| | | | | | | | t = 3.5 | |
| MCB-CM (Mittal and Jain, 2012) | 241 | 0.01 | 0.0252 | 0.0994 | 0.0151 | 0.0549 | 0.0117 | 0.0486 |
| $\beta = 0.5$ (Xie et al., 2008) | 121 | 0.01 | 0.38421 | 1.34728 | 0.49135 | 1.55470 | 0.525855 | 1.52196 |
| $\beta = 1$ (Xie et al., 2008) | 121 | 0.01 | 3.08966 | 10.4040 | 2.72048 | 8.29747 | 2.12110 | 5.94321 |
| MCB-DQM (Arora and Singh, 2013) | 121 | 0.01 | 0.00191 | 0.00777 | 0.00778 | 0.00275 | 0.006177 | 0.04335 |
| Present | 121 | 0.01 | 0.00101 | 0.00484 | 0.00060 | 0.00179 | 0.006162 | 0.04317 |

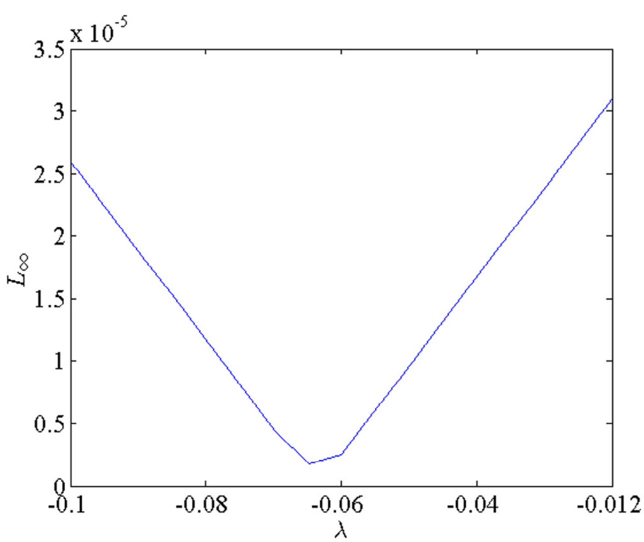


**Fig. 4 – $L_\infty$ error norm vs. $\lambda$.**

Mittal and Jain (2012), Salas (2010), Xie et al. (2008), Zhang et al. (2010), Korkmaz and Dag (2013). It is found that the present results are much better than almost all earlier methods. Physical behavior of the EMCB-DQM numeric solutions for $\upsilon = 0.005$ at different time levels with $h = 0.01$, $\Delta t = 0.01$ are shown in Fig. 5. The absolute errors for different time levels are depicted in Fig. 6. Further, it is observed that the absolute errors are much better than that given by Arora and Singh (2013).

**Table 3 – Comparison of $L_2$ and $L_\infty$ errors obtained by present method for $\upsilon = 0.005$ with the errors obtained by Korkmaz and Dag (2013) and Arora and Singh (2013) at $t = 3.6$.**

| | Present | Arora and Singh (2013) | Korkmaz and Dag (2013) | | |
|---|---|---|---|---|---|
| | | | Method I | Method II | Method II |
| $L_{2\times10^3}$ | 0.01 | 0.01 | 0.18 | 0.16 | 0.14 |
| $L_{\infty\times10^3}$ | 0.07 | 0.07 | 0.46 | 0.52 | 0.54 |

Example 2. In this example, the particular solution of the Burgers' equation (1.1) is taken for $\alpha = 1$ over the region [0,2] as considered by Arora and Singh (2013)

$$u(x,t) = 2\pi\upsilon \frac{\sin(\pi x)\exp(-\pi^2\upsilon^2 t) + 4\sin(2\pi x)\exp(-4\pi^2\upsilon^2 t)}{4 + \cos(\pi x)\exp(-\pi^2\upsilon^2 t) + 2\cos(2\pi x)\exp(-4\pi^2\upsilon^2 t)},$$
$$t \geq 0,$$

where the initial and boundary conditions are extracted from the analytic solution.

For the Ex. 2, the comparison of $L_2$ and $L_\infty$ errors with parameters $h = 0.1$, $\Delta t = 0.01$, $\lambda = 0.9$ at $t = 1$ for different values of $\upsilon$ is reported in Table 4. It is evident that obtained result by the EMCB-DQM is better than obtained by Mittal and Jain (2012) and Arora and Singh (2013). Also, it is observed that error is decreasing rapidly as $\upsilon$ increases. The physical behavior of the EMCB-DQM numeric solutions at different time levels are depicted in Fig. 7.

Example 3. Consider the analytical solutions of Eqs. (1.4) and (1.5), as generated by Fletcher (1983)

**Table 4 – Comparison of obtained $L_2$ and $L_\infty$ errors with the errors obtained in (Arora and Singh, 2013; Mittal and Jain, 2012).**

| $\upsilon$ | Mittal and Jain (2012) $h = 0.025$, $\Delta t = 10^{-3}$ | | Arora and Singh (2013) $h = 0.1$, $\Delta t = 0.01$ | | EMCB-DQM $h = 0.1$, $\Delta t = 0.01$ | |
|---|---|---|---|---|---|---|
| | $L_2$ | $L_\infty$ | $L_2$ | $L_\infty$ | $L_2$ | $L_\infty$ |
| $10^{-2}$ | 3.13E-02 | 2.66E-02 | 2.92E-02 | 2.63E-02 | 2.87E-02 | 2.58E-02 |
| $10^{-3}$ | 4.45E-04 | 3.59E-04 | 3.93E-04 | 3.45E-04 | 3.33E-04 | 3.75E-04 |
| $10^{-4}$ | 4.61E-06 | 3.72E-06 | 4.09E-06 | 3.55E-06 | 3.42E-06 | 3.88E-06 |
| $10^{-5}$ | 4.62E-08 | 3.74E-08 | 4.11E-08 | 3.56E-08 | 3.43E-08 | 3.89E-08 |
| $10^{-6}$ | 4.62E-10 | 3.74E-10 | 4.11E-10 | 3.56E-10 | 3.43E-10 | 3.89E-10 |

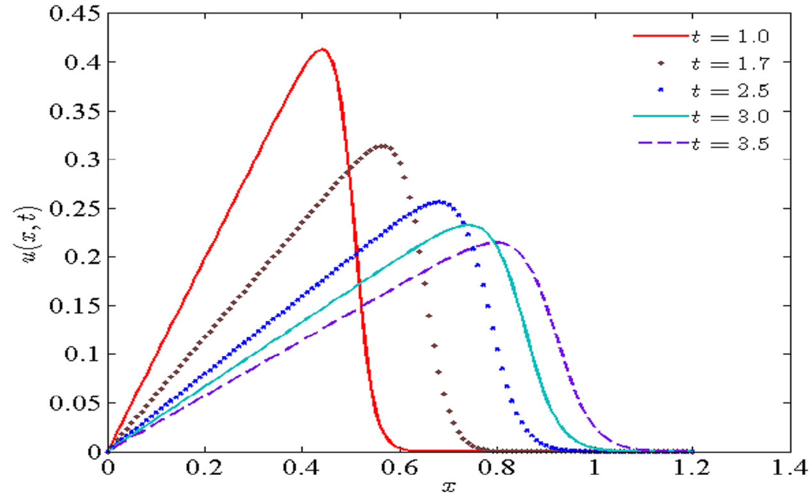


**Fig. 5 – Physical behavior of the EMCB-DQM numeric solutions of Ex. 1 for $\upsilon = 0.005$ at different time levels with $h = 0.01$, $\Delta t = 0.01$.**

$$\left.\begin{aligned} u(x,y,t) &= \frac{3}{4} - \frac{1}{4[1+\exp((-4x+4y-t)Re/32)]} \\ v(x,y,t) &= \frac{3}{4} + \frac{1}{4[1+\exp((-4x+4y-t)Re/32)]} \end{aligned}\right\}; a_1 \le x \le a_2, \quad b_1 \le y \le b_2, \tag{5.3}$$

The domain $0 \le x \le 1, 0 \le y \le 1$ is considered as the computational domain, and the initial and boundary conditions are extracted from the analytical solutions (5.3). The numerical solution is computed with the parameters: $\upsilon = 10^{-2}$, $\lambda = 0.4$, $\Delta t = 0.0001$ at $t = 1.0$ for different grid sizes and is shown in Table 5 and Table 6 in terms of $L_2$ and $L_\infty$ errors for $u$ and $v$ components, respectively. The rate of convergence (ROC) is also shown. It is found that the EMCB-DQM performs much better than Srivastava et al. (2013) and Shukla et al. (2014) and gives more than quadratic rate of convergence. Computed EMCB-DQM solutions of $u$ and $v$ components for $\upsilon = 10^{-2}$ and $\lambda = 0.4$ at $t = 0.5$ are depicted in Fig. 8 while Fig. 9 shows analytical solutions of $u$ and $v$ components, respectively.

Example 4. We consider one dimensional coupled Burgers' equation (Doha et al., 2014; Mittal and Arora, 2011)

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - 2u\frac{\partial u}{\partial x} + \frac{\partial (uv)}{\partial x} = 0, \tag{5.4}$$

$$\frac{\partial v}{\partial t} - \frac{\partial^2 v}{\partial x^2} - 2v\frac{\partial v}{\partial x} + \frac{\partial (uv)}{\partial x} = 0, \tag{5.5}$$

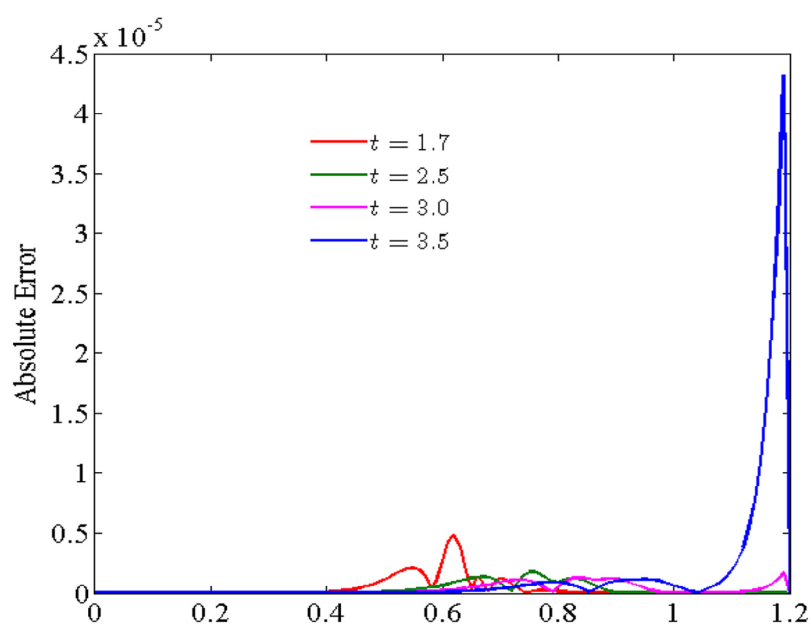


**Fig. 6 – Absolute errors in the EMCB-DQM numeric solutions of Ex. 1 for $\upsilon = 0.005$ at different time levels with $h = 0.01$, $\Delta t = 0.01$.**

**Table 5 – Errors and rate of convergence for $u$-component for $\upsilon = 10^{-2}$, $\Delta t = 0.0001$ at $t = 1.0$.**

| Grid | $L_2$ | | | | $L_\infty$ | | | |
|---|---|---|---|---|---|---|---|---|
| | Srivastava et al. (2013) | Shukla et al. (2014) | EMCB-DQM | | Srivastava et al. (2013) | Shukla et al. (2014) | EMCB-DQM | |
| | | | $\lambda = 0.4$ | ROC | | | $\lambda = 0.4$ | ROC |
| 4 × 4 | 8.5708E-02 | 1.6388E-02 | 1.5787E-02 | – | 9.7046E-02 | 2.8788E-03 | 2.5538E-03 | |
| 8 × 8 | 4.9429E-02 | 1.9286E-03 | 1.7600E-03 | 3.165 | 4.6886E-02 | 1.9572E-04 | 1.7765E-04 | 3.846 |
| 16 × 16 | 1.9192E-02 | 3.9474E-04 | 3.2260E-04 | 2.448 | 2.0467E-02 | 2.0486E-05 | 1.9610E-05 | 3.192 |
| 32 × 32 | 8.6812E-03 | 8.1181E-05 | 6.2990E-05 | 2.357 | 9.0744E-03 | 2.2202E-06 | 1.9445E-06 | 3.322 |
| 64 × 64 | – | 1.5322E-05 | 1.1280E-05 | 2.481 | – | 2.1838E-07 | 1.7271E-07 | 3.493 |

**Table 6 – Errors and rate of convergence for $\upsilon$-component for $\upsilon = 10^{-2}$, $\Delta t$ =0.0001 at $t = 1.0$.**

| Grid | $L_2$ | | | | $L_\infty$ | | | |
|---|---|---|---|---|---|---|---|---|
| | Srivastava et al. (2013) | Shukla et al. (2014) | EMCB-DQM | | Srivastava et al. (2013) | Shukla et al. (2014) | EMCB-DQM | |
| | | | $\lambda = 0.4$ | ROC | | | $\lambda = 0.4$ | ROC |
| 4 × 4 | 8.5708E-02 | 1.6388E-02 | 1.5787E-02 | - | 9.7046E-02 | 2.8788E-03 | 2.5538E-03 | - |
| 8 × 8 | 4.9431E-02 | 1.9286E-03 | 1.7600E-03 | 3.165 | 4.6887E-02 | 1.9573E-04 | 1.7765E-04 | 3.846 |
| 16 × 16 | 1.9196E-02 | 3.9474E-04 | 3.2260E-04 | 2.448 | 2.0471E-02 | 2.0486E-05 | 1.9610E-04 | 3.192 |
| 32 × 32 | 8.6878E-03 | 8.1181E-05 | 6.2990E-05 | 2.357 | 9.0813E-03 | 2.2202E-06 | 1.9445E-06 | 3.322 |
| 64 × 64 | – | 1.5322E-05 | 1.1280E-05 | 2.481 | – | 2.1838E-07 | 1.7271E-07 | 3.493 |

**Table 7 – Maximum absolute error for Ex. 4.**

| N | Present for $\lambda = -0.37$ | | Doha et al. (2014) for (–1/2, –1/2) | |
|---|---|---|---|---|
| | $u$-component | $\upsilon$-component | $u$-component | $\upsilon$-component |
| 4 | 2.18E-02 | 2.18E-02 | 11.87E-02 | 11.87E-02 |
| 8 | 8.66E-04 | 8.66E-04 | 3.44E-04 | 3.44E-04 |

together with the initial and boundary conditions

$$u(x,0) = \upsilon(x,0) = \sin(x),$$

$$u(-\pi,t) = \upsilon(-\pi,t) = u(\pi,t) = \upsilon(\pi,t) = 0.$$

The numerical solution of Ex. 4 is obtained for different values of $t$, Grid size $N$ for free parameter $\lambda = -0.37$ and compared with the solution obtained by Mittal and Arora (2011) and Doha et al. (2014) in terms of maximum absolute error. Table 7 and Table 8 shows that the present results are much better than those obtained by Mittal and Arora (2011) but in comparison
with Doha *et al.*, the results are in good agreement with small $N$. So this method can be considered as an alternate approach to solve the partial differential equations.

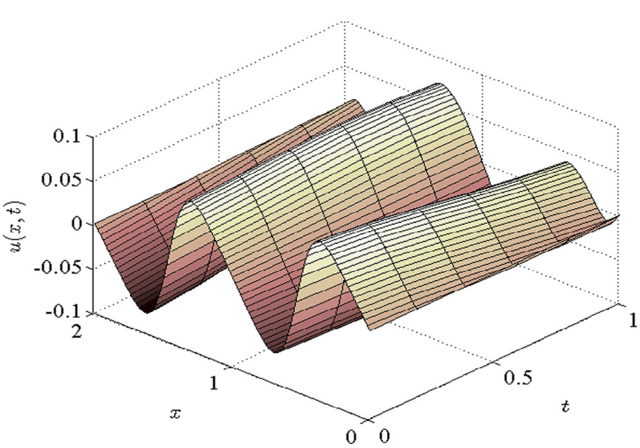


**Fig. 7 – Physical behavior of the EMCB-DQM numeric solutions of Ex. 2 for $\upsilon = 0.01$ at different time levels with $h = 0.02$, $\Delta t = 0.01$.**

## 6. Conclusions

In this paper, a new numerical method is set up for solving the nonlinear partial differential equations. The proposed method tested for one and two dimensional Burgers' equations.

**Table 8 – Maximum absolute error for Ex. 4.**

| t | Present for $\lambda = -0.37$ and N = 200 | Mittal and Arora (2011) N = 400 | Doha et al. (2014) for (–1/2, –1/2) and N = 20 |
|---|---|---|---|
| 0.1 | 6.88E-07 | 1.86E-06 | 1.36E-08 |
| 0.5 | 2.27E-06 | 6.22E-06 | 1.76E-08 |
| 1.0 | 2.72E-06 | 7.56E-06 | 2.37E-08 |

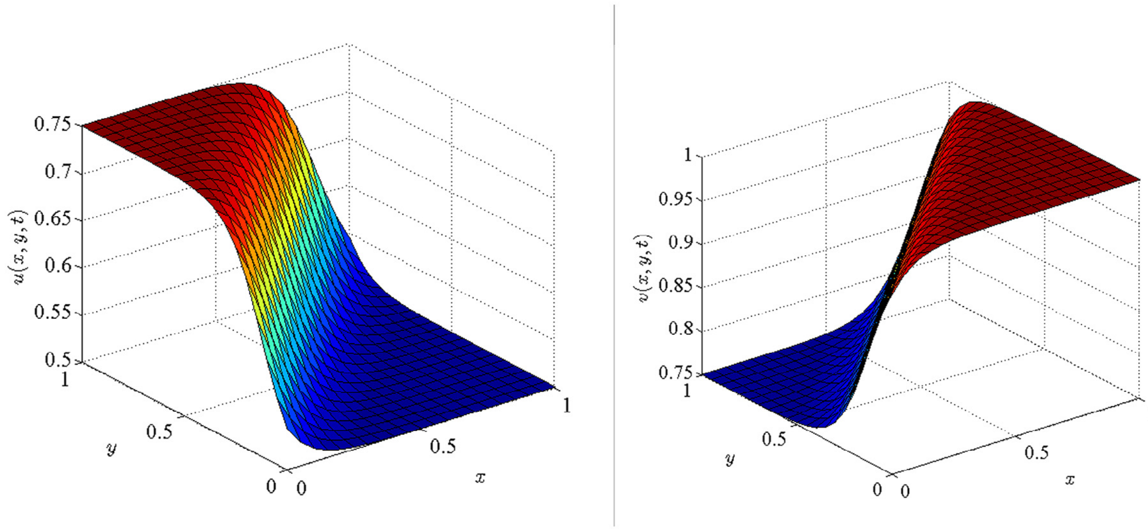


**Fig. 8 – Numerical solution of Ex. 3 at $t = 0.5$ with $h = 0.05$, $\Delta t = 0.0001$ and $\upsilon = 10^{-2}$**

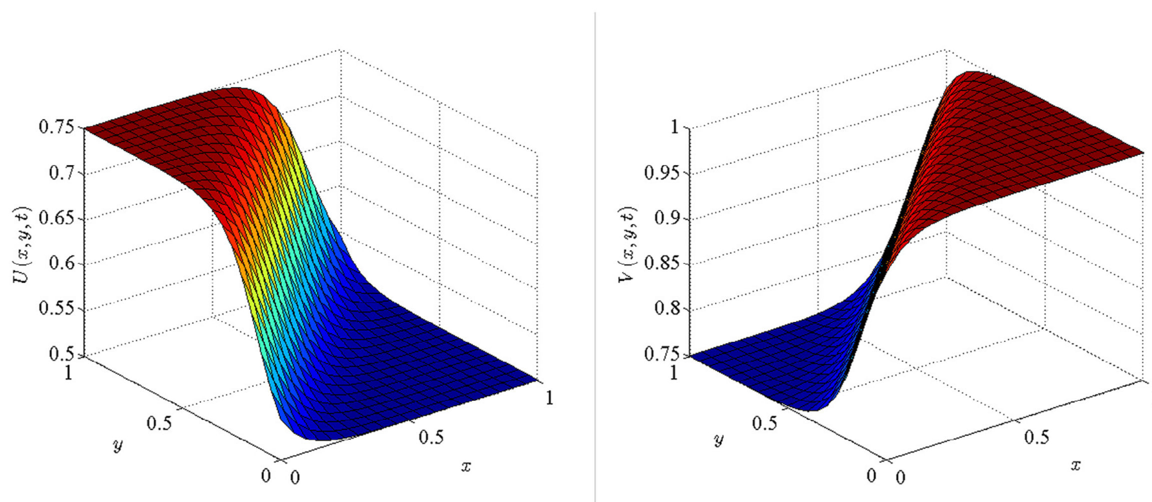


**Fig. 9 – Exact solution of Ex. 3 at $t = 0.5$ with $h = 0.05$, $\Delta t = 0.0001$ and $\upsilon = 10^{-2}$.**

Finally, the present analysis summarizes the following outcomes:

- (i) The cost of the proposed algorithm is the same as the modified cubic B-spline differential quadrature method but the errors are less than later one.
- (ii) The proposed method gives better results than results obtained by Dag et al. (2005), Xie et al. (2008), Chung et al. (2010), Salas (2010), Zhang et al. (2010), Korkmaz et al. (2011), Mittal and Jain (2012), Korkmaz and Dag (2013), Arora and Singh (2013), Srivastava et al. (2013), Shukla et al. (2014), Mittal and Arora (2011).
- (iii) To the best knowledge of the author, this is a new differential quadrature technique for solving differential equations.
- (iv) The low memory storage and easiness of the implementation can be counted as main advantage of the algorithm.

## REFERENCES

Aksan EN. Quadratic B-spline finite element method for numerical solution of the Burgers' equation. Appl Math Comput 2006;174:884–96.

Arora G, Singh BK. Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method. Appl Math Comput 2013;224(1):166–77.

Bateman H. Some recent researches on the motion of fluids. Mon Weather Rev 1915;43:163–70.

Bellman R, Kashef BG, Casti J. Differential quadrature: a technique for the rapid solution of nonlinear differential equations. J Comput Phy 1972;10:40–52.

Bhrawy AH, Zaky MA. Numerical algorithm for the variable-order Caputo fractional functional differential equation. Nonlinear Dyn 2016;85:1815–1823.

Bhrawy AH, Zaky MA, Baleanu D. New numerical approximations for space-time fractional Burgers' equations via a Legendre

spectral-collocation method. Romanian Reports of Physics 2015;67:340–9.
Burgers JM. Mathematical example illustrating relations occurring in the theory of turbulent fluid motion. Trans Roy Neth Acad Sci Amsterdam 1939;281–334.
Cecchi MM, Nociforo R, Grego PP. Space-time finite elements numerical solution of Burgers problems. Le Matematiche 1996;51(1):43–57.
Chung J, Kim E, Kim Y. Asymptotic agreement of moments and higher order contraction in the Burgers equation. J Diff Equ 2010;248(10):2417–34.
Cole JD. On a quasilinear parabolic equations occurring in aerodynamics. Quart Appl Math 1951;9:225–36.
Dag I, Irk D, Sahin A. B-spline collocation methods for numerical solutions of the Burgers' equation. Math Probl Eng 2005;5:521–38.
Dag I, Irk D, Sari M. The extended cubic B-spline algorithm for a modified regularized long wave equation. Chin Phys B 2013;22(4):040207.
Dogan A, Galerkin A. Finite element approach to Burgers' equation. Appl Math Comput 2004;157:331–46.
Doha EH, Bhrawy AH, Abdelkawy MA, Hafez RM. A Jacobi collocation approximation for nonlinear coupled viscous Burgers' equation. Cent Eur J Phys 2014;12(2):111–22.
Esen A, Tasbozan O. Numerical solution of time fractional Burgers equation by cubic B-spline finite elements. Mediterr J Math 2016;13(3):1325–37.
Esipov SE. Coupled Burgers' equations: a model of poly-dispersive sedimentation. Phys Rev 1995;52:3711–18.
Fletcher CAJ. Generating exact solutions of the two-dimensional Burgers' equation. Int J Numer Meth Fluids 1983;3:213–16.
Gottlieb S, Ketcheson DI, Shu CW. High order strong stability preserving time discretizations. J Sci Comput 2009;38:251–89.
Hassanien IA, Salama AA, Hosham HA. Fourth-order finite difference method for solving Burgers' equation. Appl Math Comput 2005;170:781–800.
Jiang Z, Wang R. An improved numerical solution of Burgers' equation by cubic B-spline Quasi-interpolation. J Inf Comput Sci 2010;7(5):1013–21.
Kadalbajoo MK, Sharma KK, Awasthi A. A parameter-uniform implicit difference scheme for solving time-dependent Burgers' equation. Appl Math Comput 2005;170:1365–93.
Korkmaz A, Dag I. Shock wave simulations using sinc differential quadrature method. Eng Comput Int J Comput Aided Eng Softw 2011;28(6):654–74.
Korkmaz A, Dag I. Cubic B-spline differential quadrature methods and stability for Burgers' equation. Eng Comput Int J Comput Aided Eng Software 2013;30(3):320–44.
Korkmaz A, Aksoy AM, Dag I. Quartic B-spline differential quadrature method. Int J Nonlinear Sci 2011;11(4):403–11.
Kutluay S, Bahadir AR, Ozdes A. Numerical solution of one-dimensional Burgers' equation: explicit and exact-explicit finite difference methods. J Comput Appl Math 1999;103:251–61.
Kutluay S, Esen A, Dag I. Numerical solutions of the Burgers' equation by the least-squares quadratic B-spline finite element method. J Comput Appl Math 2004;167:21–33.
Liao W. An implicit fourth-order compact finite difference scheme for one-dimensional Burgers' equation. Appl Math Comput 2008;206:755–64.
Mittal RC, Arora G. Numerical solution of the coupled viscous Burgers' equation. Commun Nonlinear Sci Numer Simul 2011;16(3):1304–13.
Mittal RC, Jain RK. Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method. Appl Math Comput 2012;218:7839–55.
Ozis T, Aksan EN, Ozdes A. A finite element approach for solution of Burgers' equation. Appl Math Comput 2003;139:417–28.
Ozis T, Esen A, Kutluay S. Numerical solution of Burgers' equation by quadratic B-spline finite elements. Appl Math Comput 2005;165:237–49.
Quan JR, Chang CT. New insights in solving distributed system equations by the quadrature methods-I. Comput Chem Eng 1989;13:779–88.
Saka B, Dag I. Quartic B-spline collocation method to the numerical solutions of the Burgers' equation. Chaos Solitons Fractals 2007;32:1125–37.
Saka B, Sahin A, Dag I. B-spline collocation algorithms for numerical solutions of the RLW equation. Num Method Partial Differ Equation 2009;27(3):581–607.
Salas AH. Symbolic computation of solutions for a forced Burgers equation. Appl Math Comput 2010;216(1):18–26.
Shu C. Differential quadrature and its application in engineering. Great Britain: Athenaeum Press Ltd.; 2000.
Shu C, Chew YT. Fourier expansion-based differential quadrature and its application to Helmholtz Eigen value problems. Commun Numer Methods Eng 1997;13(8):643–53.
Shukla HS, Tamsir M, Srivastava VK, Kumar J. Numerical solution of two dimensional coupled viscous Burger equation using modified cubic B-spline differential quadrature method. AIP Advances 2014;4:117134.
Srivastava VK, Awasthi MK, Singh S. An implicit logarithmic finite difference technique for two dimensional coupled viscous Burgers' equation. AIP Advances 2013;3:122105.
Xie SS, Heo S, Kim S, Woo G, Yi S. Numerical solution of one-dimensional Burgers' equation using reproducing kernel function. J Comput Appl Math 2008;214:417–34.
Zhang L, Ouyang J, Wang X, Zhang X. Variational multiscale element-free Galerkin method for 2D Burgers equation. J Comput Phy 2010;229(19):7147–61.