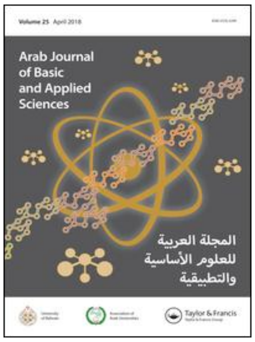

Arab Journal of Basic and Applied Sciences

ISSN: (Print) 2576-5299 (Online) Journal homepage: https://www.tandfonline.com/loi/tabs20

Solving one- and two-dimensional unsteady Burgers' equation using fully implicit finite difference schemes

N. A. Mohamed

To cite this article: N. A. Mohamed (2019) Solving one- and two-dimensional unsteady Burgers' equation using fully implicit finite difference schemes, Arab Journal of Basic and Applied Sciences, 26:1, 254-268, DOI: 10.1080/25765299.2019.1613746

To link to this article: https://doi.org/10.1080/25765299.2019.1613746

ARAB JOURNAL OF BASIC AND APPLIED SCIENCES
2019, VOL. 26, NO. 1, 254–268
https://doi.org/10.1080/25765299.2019.1613746

University of Bahrain
Taylor & Francis
Taylor & Francis Group

ORIGINAL ARTICLE

OPEN ACCESS

Check for updates

Solving one- and two-dimensional unsteady Burgers' equation using fully implicit finite difference schemes

N. A. Mohamed

Department of Engineering Mathematics and Physics, Faculty of Engineering, Zagazig University, Zagazig, Egypt

ABSTRACT

This paper introduces new fully implicit numerical schemes for solving 1D and 2D unsteady Burgers' equation. The non-linear Burgers' equation is discretized in the spatial direction by using second order Finite difference method which converts the Burgers' equation to non-linear system of ODEs. Then, the backward differentiation formula of order two (BDF-2) is employed to march the solution in the time direction. The non-linear term in the obtained system is linearized without any transformation; and it forms a system of linear algebraic equations that is solved by using Thomas's algorithm. Accuracy and performance of the proposed schemes are studied by using four test problems with Dirichlet and Neumann boundary conditions. Comparing the numerical results with exact solutions and the solutions of other schemes shows that the proposed schemes are simple, efficient and accurate even for the cases with high Reynolds numbers.

ARTICLE HISTORY
Received 18 September 2018
Revised 15 April 2019
Accepted 27 April 2019

KEYWORDS
Burgers equation; finite difference method; nonlinear analysis; Reynolds number; error analysis

1. Introduction

Non-linear partial differential equations are found in various fields of science and engineering as in plasma fields (Seadawy, 2015a, 2015b, 2017a, 2017b), nanofluid flow (Sheikholeslami, Zeeshan, & Majeed, 2018; Sheikholeslami & Zeeshan, 2018; Sheikholeslami & Zeeshan, 2017) and shallow-water waves (Seadawy, 2017c). Burgers' equation is considered one of the most common non-linear partial differential equations. It arises in many physical problems, including shock flows, traffic flows, non-linear wave propagation, waves of impact and sound waves in a viscous medium.

This paper considers two cases of the unsteady Burgers' equation:

1.1. One-dimensional Burgers' equation

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}=\nu\frac{\partial^2 u}{\partial x^2},\quad P_1\le x\le P_2,\quad t\in[0,\ T],\tag{1}$$

with the initial condition

$$u(x,\ 0)=u_0(x),\tag{2}$$

and, the Dirichlet boundary conditions

$$u(P_1,\ t)=u(P_2,\ t)=0;\quad t\in[0,\ T],\tag{3a}$$

or, the Mixed boundary conditions

$$u_x(P_1,\ t)=0,\ \ u(P_2,\ t)=f(x,t);\quad t\in[0,\ T],\tag{3b}$$

1.2 Two-dimensional Burgers' equation

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+u\frac{\partial u}{\partial y}=\nu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right),\tag{4}$$
$$z_1\le x\le z_2,\ \ z_3\le y\le z_4,\ \ t\in[0,\ T],$$

with the initial condition

$$u(x,y,\ 0)=u_0(x,y),\tag{5}$$

and the Dirichlet boundary conditions

$$\begin{aligned}&u(z_1,\ y,t)=g_1(y,t),\ \ u(z_2,\ y,t)=g_2(y,t),\\&u(x,\ z_3,t)=g_3(x,t),\\&u(x,\ z_4,t)=g_4(x,t);\ \ t\in[0,\ T],\end{aligned}\tag{6}$$

where u is the velocity, $\nu=1/Re>0$ is an arbitrary number which is known as kinematic viscosity coefficient, Re is the Reynolds number, $u_0,\ g_1,\ g_2,\ g_3,\ g_4$ are known functions, $\frac{\partial u}{\partial t}$ is unsteady term, $u\frac{\partial u}{\partial x}$ is the non-linear convective term, and $\nu\frac{\partial^2 u}{\partial x^2}$ is the diffusive term. Burgers' equation is taken as a model for testing numerical methods; and also for obtaining the numerical solutions of the equation for small viscosity values due to its similarity to the Navier-Stokes equations. Scientists and engineers are interested in

CONTACT N. A. Mohamed norhan.alla@gmail.com

using various numerical techniques in order to study the properties of the Burgers' equation due to its applicability in the different fields of science and engineering.

Bateman in (Bateman 1915) is the first who introduced Burgers' equation. Later, in the papers (Burgers 1939, 1948), Burgers investigated various aspects of turbulence and provided a mathematical model illustrating the theory. Also, he studied the statistical and spectral aspects of the equation and related systems of equations. Hopf and Cole in (Cole 1951, Hopf 1950) studied the generalproperties of the equation in order to show the typical characteristics of the shock wave theory.

The solution of the Burgers' equation is an active area through which researchers developed many numerical algorithms; and they obtained its approximate solution. These numerical algorithms depended on numerical methods such as finite difference method (Ciment, Leventhal, & Weinberg, 1978; Iskandar & Mohsen, 1992), explicit and exact explicit finite difference methods (Kutulay, Bahadir, & Odes, 1999), fourth order finite difference method (Hassanien, Salama, & Hosham, 2005), higher-order accurate finite difference method (Zhanlav, Chuluunbaatar, & Ulziibayar, 2015), collection of numerical techniques based on finite difference (Mukundan & Awasthi, 2015; Radwan, 2005), finite elements method (Dogan, 2004; Ozis, Aksan, & Ozdes, 2003), quadratic B-splines finite element method (Aksan, 2006), spectral least-squares method (Heinrichs, 2007; Maerschalck & Gerritsma, 2005; Maerschalck & Gerritsma, 2008), variational iteration method (Abdou & Soliman, 2005; Biazar & Aminikhah, 2009), Adomian–Pade technique (Dehghan, Hamidi, & Shakourifar, 2007), homotopy analysis method (Rashidi, Domairry, & Dinarvand, 2009), differential transform method and the homotopy analysis method (Rashidi & Erfani, 2009), automatic differentiation method (Asaithambi, 2010), cubic spline quasi-interpolant scheme (Xu, Wang, Zhang, & Fang, 2011), Laplace decomposition method (Khan, 2014), B-splines collocation method (Ali, Gardner, & Gardner, 1992), Quartic B-spline collocation method (Saka & Dağ, 2007), Spectral collocation method (Khalifa, Noor, & Noor, 2011; Khater, Temsah, & Hassan, 2008), Cubic Hermite collocation method (Ganaie & Kukreja, 2014), Sinc differential quadrature method (Korkmaz & Dağ, 2011a), Polynomial based differential quadrature method (Korkmaz & Dağ, 2011b), Modified cubic B-spline differential quadrature method (Arora & Singh, 2013; Mittal & Jain, 2012), Exponential modified cubic B-spline differential quadrature method (Tamsir, Srivastava, & Jiwari, 2016), Hybrid numerical scheme based on Haar wavelets (Jiwari, 2015), High order splitting method (Seydaoğlu, Erdoğan, & Öziş, 2016), etc.

It is known that the big challenge in solving Burgers' equation occurs when it involves high Reynolds numbers due to the formation of a thin layer that is close to the wall that is known as the boundary layer where the velocity transition from the limiting solution's finite value is close to the wall to the value of zero directly at the wall. So, at large Reynolds numbers, the viscosity must be taken into account in order to satisfy the no-slip condition. The concept of the boundary layer implies that flows at high Reynolds numbers can be divided up into two unequally large regions where the second region is the very thin boundary layer at the wall where the viscosity must be taken into account (Kadalbajoo & Gupta, 2010). Different numerical methods considered this problem and obtained an accurate solution as in (Arafa, 1996). It used Finite difference method to obtain solution for $Re = 10,000$, Zhang et al. as in (Zhang, Wei, Kouri, & Hoffman, 1997). It solved numerically Burgers' equation by using a new approach based on the distributed approximating function for $Re = 10^{-5}$ with a very large mapping parameter to shift most of the 200 grid points into the boundary layer region to obtain oscillation-free solution and small time increment is chosen in order to ensure high accuracy. Liu (Liu, 2006) used a group of preserving schemes to solve Burgers' equation for $Re = 10000,\ 20000$. An exponentially fitted method (Ozis & Erdogan, 2009) succeeded in solving non-linear problems of Burgers' type at $Re = 10^5,\ 10^6$. Liu (Liu, 2009) used a new fictitious time integration method and the solution of $Re = 10000$ was obtained.

In this paper, the proposed schemes based on second order finite difference method are used to solve the one- and two- dimensional Burgers' equation; and the linearization is done as in (Kay, Gresho, Griffiths, & Silvester, 2010). The results are compared with exact solution and other techniques even for the high Reynolds numbers.

The structure of the paper is as follows. In Section 2, the solution procedure is described. In Section 3, error analysis of the proposed scheme is presented. Numerical results of test problems and discussions are listed in Section 4. Finally, conclusions are given in Section 5.

2. The solution procedure

The following is the procedure illustrating two cases that are related to the 1-D Burgers' and 2-D Burgers' equations:

2.1. 1-D Burgers' equation

The Burgers' Eq. (1) can be rewritten as:

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + \nu\frac{\partial^2 u}{\partial x^2},\ P_1 \le x \le P_2,\ t \in [0,\ T], \quad (7)$$

with initial and boundary conditions as in (2), (3), respectively. Discretization is obtained by using central finite difference discretization along x direction while t direction is discretized by using p^{th} order backward differentiation formula (BDF) (Atkinson, Han, & Stewart, 1999).

Divide the interval $[0,\ T]$ into N steps $0 = t_0 \le t_1 \le \ldots \le t_N = T$, with constant time step $\Delta t = T/N$ and $t_n = n*\Delta t$ for $n = 1, 2, \ldots, N$ and apply second-order backward differentiation formula (BDF-2) (Atkinson et al., 1999) along t direction in Eq. (7); it becomes:

$$u^{n+1} = \frac{4}{3}u^n - \frac{1}{3}u^{n-1} + \frac{2}{3}\Delta t\ \left(\nu\ u_{xx}^{n+1} - u^{n+1}u_x^{n+1}\right) \quad (8)$$

Applying Eq. (8), at every time step, needs to get the solutions at the previous time levels $n-1$ and n; so:

1. For $n = 1$, the solution at time level $n-1$ is the solution that is obtained from initial condition and the solution at time level n in this work is obtained by using backward Euler Formula (BDF-1) (Atkinson et al., 1999):
$$u^n = u^{n-1} + \ \Delta t\ \left(\nu\ u_{xx}^{n-1} - u^{n-1}u_x^{n-1}\right)$$
2. For $n > 1$, the solutions at the previous time levels $n-1$ and n are already obtained and the solution at the time level $n+1$ is obtained by using Eq. (8).

Also, from Eq. (8), the non-linear term $u^{n+1}u_x^{n+1}$ is needed to be computed at every time step. In this work the linearization of this term is done as in (Kay et al., 2010) such that $u^{n+1}u_x^{n+1} \approx w^{n+1}u_x^{n+1}$, where w^{n+1} is computed by linear extrapolation using u^n and u^{n-1} as

$$u^{n+1} \cong w^{n+1} = \left(1 + \left(\frac{K_{n+1}}{K_n}\right)\right)u^n - \left(\frac{K_{n+1}}{K_n}\right)u^{n-1} \quad (9)$$

where $K_{n+1} = t_{n+1} - t_n$ and $K_n = t_n - t_{n-1}$. Substitution of Eq. (9) in Eq. (8) yields,

$$u^{n+1} = \frac{4}{3}u^n - \frac{1}{3}u^{n-1} + \frac{2}{3}\Delta t\ \Big(\nu\ u_{xx}^{n+1} - \left(\left(1 + \left(\frac{K_{n+1}}{K_n}\right)\right)u^n - \left(\frac{K_{n+1}}{K_n}\right)u^{n-1}\right)u_x^{n+1}\Big) \quad (10)$$

Along x direction, the central finite difference discretization for the terms u_{xx}^{n+1} and u_x^{n+1} in Eq. (10) is used within one of the following schemes.

2.1.1. Uniform spacing scheme

Divide the interval $[P_1,\ P_2]$ into M nodes $P_1 = x_0 \le x_1 \le \ldots \le x_M = P_2$, with constant spacing step $h = (P_2 - P_1)/M$ and $x_i = P_1 + ih$ for $i = 1, 2, \ldots,$ M, given u^n at time level t_n and compute u^{n+1} at t_{n+1} via:

$$u_i^{n+1} = \frac{4}{3}u_i^n - \frac{1}{3}u_i^{n-1} + \frac{2}{3}\Delta t\ \Big(\frac{\nu}{h^2}\left[u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}\right] - \frac{1}{2h}\left(\left(1 + \left(\frac{K_{n+1}}{K_n}\right)\right)u_i^n - \left(\frac{K_{n+1}}{K_n}\right)u_i^{n-1}\right)\left[u_{i+1}^{n+1} - u_{i-1}^{n+1}\right]\Big) \quad (11)$$

Since the constant time step is applied so $K_{n+1} = K_n = \Delta t$ and Eq. (11) will be written as:

$$u_i^{n+1} = \frac{4}{3}u_i^n - \frac{1}{3}u_i^{n-1} + \frac{2}{3}\Delta t\ \Big(\frac{\nu}{h^2}\left[u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}\right] - \frac{1}{2h}\left(2u_i^n - u_i^{n-1}\right)\left[u_{i+1}^{n+1} - u_{i-1}^{n+1}\right]\Big) \quad (12)$$

Rearrange Eq. (12) to get the following form:

$$\left[1 + \frac{4}{3}\frac{\nu}{h^2}\Delta t\right]u_i^{n+1} - \frac{2}{3}\Delta t\left[\frac{\nu}{h^2} - \frac{\left(2u_i^n - u_i^{n-1}\right)}{2h}\right]u_{i+1}^{n+1} - \frac{2}{3}\Delta t\left[\frac{\nu}{h^2} + \frac{\left(2u_i^n - u_i^{n-1}\right)}{2h}\right]u_{i-1}^{n+1} = \frac{4}{3}u_i^n - \frac{1}{3}u_i^{n-1} \quad (13)$$

Eq. (13) can be written in a simple form as:

$$\alpha_i u_i^{n+1} + \beta_i u_{i+1}^{n+1} + \gamma_i u_{i-1}^{n+1} = f_i \quad (14)$$

where,

$$\left.\begin{aligned} \alpha_i &= \left[1 + \frac{4}{3}\frac{\nu}{h^2}\Delta t\right],\ \beta_i = -\frac{2}{3}\Delta t\left[\frac{\nu}{h^2} - \frac{\left(2u_i^n - u_i^{n-1}\right)}{2h}\right] \\ \gamma_i &= -\frac{2}{3}\Delta t\left[\frac{\nu}{h^2} + \frac{\left(2u_i^n - u_i^{n-1}\right)}{2h}\right],\ f_i = \frac{4}{3}u_i^n - \frac{1}{3}u_i^{n-1} \end{aligned}\right\}$$

Applying Eq. (14) at each point i results in a system of tri-diagonal matrix equation $Bu = F$ that can be solved efficiently by Thomas algorithm.

2.1.2. Non-uniform spacing scheme

Divide the interval $[P_1,\ P_2]$ into two intervals $[P_1,\ P_3]$ and $[P_3,\ P_2]$, where $P_3 \in [P_1,\ P_2]$. l_1 is the length of the interval $[P_1,\ P_3]$ that is divided with constant spacing step size h then the number of steps $M_1 = \frac{P_3 - P_1}{h}$ where $P_1 = x_0 \le x_1 \le \ldots \le x_{M_1} = P_3$ and $x_i = P_1 + ih$ for $i = 1, 2, \ldots, M_1$ while l_2 is the length of the interval $[P_3,\ P_2]$ that is divided with constant spacing step size $r = \varepsilon h$ then the number of steps $M_2 = \frac{P_2 - P_3}{r}$ where $P_3 = x_{M_1} \le \ldots \le x_i \le \ldots \le x_{M_2} = P_2$ and $x_i = P_3 + ir$ for $i = 1, 2, \ldots, M_2$. Taylor series expansion is applied to get $\frac{\partial u}{\partial x}$ and $\frac{\partial^2 u}{\partial x^2}$ with unequal space size as follows:

$$\frac{\partial u_i}{\partial x}=\frac{1}{h}\left(\frac{1}{\varepsilon(1+\varepsilon)}u_{i+1}-\left(\frac{1-\varepsilon}{\varepsilon}\right)u_i-\left(\frac{\varepsilon}{1+\varepsilon}\right)u_{i-1}\right) \tag{15}$$

and,

$$\frac{\partial^2 u_i}{\partial x^2}=\frac{2}{h^2}\left(\frac{1}{\varepsilon(1+\varepsilon)}u_{i+1}-\left(\frac{1}{\varepsilon}\right)u_i-\left(\frac{1}{1+\varepsilon}\right)u_{i-1}\right) \tag{16}$$

Substitute with Eqs. (15) (16) in Eq. (10) to get:

$$\begin{aligned}u_i^{n+1}=&\frac{4}{3}u^n-\frac{1}{3}u^{n-1}+\frac{2}{3}\Delta t\\ &\nu\left(\frac{2}{h^2}\left(\frac{1}{\varepsilon(1+\varepsilon)}u_{i+1}^{n+1}-\left(\frac{1}{\varepsilon}\right)u_i^{n+1}-\left(\frac{1}{1+\varepsilon}\right)u_{i-1}^{n+1}\right)\right)\\ &-\left(\left(1+\left(\frac{K_{n+1}}{K_n}\right)\right)u_i^n-\left(\frac{K_{n+1}}{K_n}\right)u_i^{n-1}\right)\\ &\frac{1}{h}\left(\frac{1}{\varepsilon(1+\varepsilon)}u_{i+1}^{n+1}-\left(\frac{1-\varepsilon}{\varepsilon}\right)u_i^{n+1}-\left(\frac{\varepsilon}{1+\varepsilon}\right)u_{i-1}\right)\end{aligned} \tag{17}$$

Equal time steps are used such that $K_n=K_{n+1}=\Delta t$, then Eq. (17) will be:

$$\begin{aligned}u_i^{n+1}=&\frac{4}{3}u^n-\frac{1}{3}u^{n-1}+\frac{2}{3}\Delta t\\ &(\frac{2\nu}{h^2}\left(\frac{1}{\varepsilon(1+\varepsilon)}u_{i+1}^{n+1}-\left(\frac{1}{\varepsilon}\right)u_i^{n+1}-\left(\frac{1}{1+\varepsilon}\right)u_{i-1}^{n+1}\right)\\ &-\left(2u_i^n-u_i^{n-1}\right)\frac{1}{h}\left(\frac{1}{\varepsilon(1+\varepsilon)}u_{i+1}^{n+1}-\left(\frac{1-\varepsilon}{\varepsilon}\right)u_i^{n+1}-\left(\frac{\varepsilon}{1+\varepsilon}\right)u_{i-1}\right))\end{aligned} \tag{18}$$

Simplifying Eq. (18) leads to:

$$\begin{aligned}&\left[1+\frac{4\nu\Delta t}{3\varepsilon h^2}-\left(\frac{2}{3}\Delta t\frac{1-\varepsilon}{h\varepsilon}\left(2u_i^n-u_i^{n-1}\right)\right)\right]u_i^{n+1}\\ &+\frac{2}{3}\Delta t\left[-\frac{2\nu}{h^2\varepsilon(1+\varepsilon)}+\left(\frac{1}{h\varepsilon(1+\varepsilon)}\left(2u_i^n-u_i^{n-1}\right)\right)\right]u_{i+1}^{n+1}\\ &+\frac{2}{3}\Delta t\left[-\frac{2\nu}{h^2(1+\varepsilon)}-\left(\frac{\varepsilon}{(1+\varepsilon)h}\left(2u_i^n-u_i^{n-1}\right)\right)\right]u_{i-1}^{n+1}\\ &=\frac{4}{3}u_i^n-\frac{1}{3}u_i^{n-1}\end{aligned} \tag{19}$$

Multiplication of Eq. (19) by $h^2\varepsilon(1+\varepsilon)$ yields:

$$\begin{aligned}&\left[h^2\varepsilon(1+\varepsilon)+\frac{4}{3}\nu\Delta t(1+\varepsilon)-\left(\frac{2}{3}h\Delta t(1-\varepsilon^2)\left(2u_i^n-u_i^{n-1}\right)\right)\right]u_i^{n+1}\\ &+\frac{2}{3}\Delta t\left[-2\nu+h\left(2u_i^n-u_i^{n-1}\right)\right]u_{i+1}^{n+1}\\ &+\frac{2}{3}\Delta t\left[-2\nu\varepsilon-\left(\varepsilon^2 h\ \left(2u_i^n-u_i^{n-1}\right)\right)\right]u_{i-1}^{n+1}\\ &=h^2\varepsilon(1+\varepsilon)\left[\frac{4}{3}u_i^n-\frac{1}{3}u_i^{n-1}\right]\end{aligned} \tag{20}$$

Eq. (20) can be written in a simple form as:

$$\delta_i u_i^{n+1}+\omega_i u_{i+1}^{n+1}+\tau_i u_{i-1}^{n+1}=S_i \tag{21}$$

where,

$$\begin{aligned}\delta_i&=h^2\varepsilon(1+\varepsilon)+\frac{4}{3}\nu\Delta t(1+\varepsilon)-\frac{2}{3}h\Delta t\left((1-\varepsilon^2)\left(2u_i^n-u_i^{n-1}\right)\right)\\ \omega_i&=-\frac{4}{3}\nu\Delta t+\frac{2}{3}h\Delta t\left(2u_i^n-u_i^{n-1}\right)\\ \tau_i&=-\frac{4}{3}\nu\varepsilon\Delta t-\frac{2}{3}h\ \Delta t\left(\varepsilon^2\left(2u_i^n-u_i^{n-1}\right)\right)\\ S_i&=h^2\varepsilon(1+\varepsilon)\left[\frac{4}{3}u_i^n-\frac{1}{3}u_i^{n-1}\right]\}\end{aligned}$$

The non-uniform spacing scheme is described by the following steps:

1. Determine l_2 (the length of the second interval) to locate the internal point P_3.
2. Determine the spacing step size h for the first interval and arbitrary value of ε to get the spacing step size $r$for the second interval.
3. Apply Eq. (14) at every point in the first interval with the spacing step size h.
4. Apply Eq. (21) at the point P_3.
5. Apply Eq. (14) at every point in the second interval with the spacing step size $r=\varepsilon h$.
6. Construct and solve the system of linear equations $Gu=S$ at each time step to find u of all points by using the Thomas algorithm.

2.2. 2-D Burgers' equation

The Burgers' Eq. (4) can be rewritten as:

$$\frac{\partial u}{\partial t}=-u\frac{\partial u}{\partial x}-u\frac{\partial u}{\partial y}+\nu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right), \tag{22}$$
$$z_1\le x\le z_2,\ z_3\le y\le z_4,\ t\in[0,\ T],$$

with initial and boundary conditions as in Eqs. (5) and (6) respectively. Applying the procedure in section 2.1 to Eq. (22), we will get the following:

$$\alpha_i u_{i,j}^{n+1}+\beta_i u_{i+1,j}^{n+1}+\gamma_i u_{i-1,j}^{n+1}+\tau_i u_{i,j+1}^{n+1}+\omega_i u_{i,j-1}^{n+1}=f_i \tag{23}$$

where,

$$\alpha_{i,j}=\left[1+\frac{4}{3}\frac{\nu}{(\Delta x)^2}\Delta t+\frac{4}{3}\frac{\nu}{(\Delta y)^2}\Delta t\right]$$

$$\beta_{i,j}=-\frac{2}{3}\Delta t\left[\frac{\nu}{(\Delta x)^2}-\frac{\left(2u_{i,j}^n-u_{i,j}^{n-1}\right)}{2(\Delta x)}\right]$$

$$\gamma_{i,j}=-\frac{2}{3}\Delta t\left[\frac{\nu}{(\Delta x)^2}+\frac{\left(2u_{i,j}^n-u_{i,j}^{n-1}\right)}{2(\Delta x)}\right]$$

$$\tau_{i,j}=-\frac{2}{3}\Delta t\left[\frac{\nu}{(\Delta y)^2}-\frac{\left(2u_{i,j}^n-u_{i,j}^{n-1}\right)}{2(\Delta y)}\right]$$

Table 1. Comparison of Proposed Scheme with (Mukundan & Awasthi, 2015) at different space points with $h = 0.01, T = 0.1$ and $\nu = 10$.

	(Mukundan & Awasthi, 2015) $\Delta t = 0.0001$ (1000 time iteration)		Proposed Scheme $\Delta t = 0.00016$ (625 time iteration)	
x	BDF-2	BDF-3	BDF-2	Exact Solution
0.1	1.600 E-05	1.598 E-05	1.598 E-05	1.598 E-05
0.2	3.043 E-05	3.039 E-05	3.040 E-05	3.040 E-05
0.3	4.188 E-05	4.183 E-05	4.184 E-05	4.184 E-05
0.4	4.923 E-05	4.918 E-05	4.918 E-05	4.919 E-05
0.5	5.177 E-05	5.171 E-05	5.171 E-05	5.172 E-05
0.6	4.923 E-05	4.918 E-05	4.918 E-05	4.919 E-05
0.7	4.188 E-05	4.183 E-05	4.184 E-05	4.184 E-05
0.8	3.043 E-05	3.039 E-05	3.040 E-05	3.040 E-05
0.9	1.600 E-05	1.598 E-05	1.598 E-05	1.598 E-05

Table 2. Comparison of Proposed Scheme with (Mukundan & Awasthi, 2015) at different space points with $h = 0.01$, $T = 0.5$ and $\nu = 1$.

	(Mukundan & Awasthi, 2015) $\Delta t = 0.001$ (500 time iteration)		Proposed Scheme $\Delta t = 0.00125$ (400 time iteration)	
x	BDF-2	BDF-3	BDF-2	Exact Solution
0.1	0.002214	0.002213	0.002213	0.002213
0.2	0.004212	0.004209	0.004210	0.004210
0.3	0.005798	0.005795	0.005796	0.005796
0.4	0.006819	0.006815	0.006816	0.006816
0.5	0.007172	0.007168	0.007170	0.007169
0.6	0.006823	0.006820	0.006821	0.006821
0.7	0.005806	0.005803	0.005804	0.005804
0.8	0.004220	0.004217	0.004218	0.004218
0.9	0.002219	0.002218	0.002218	0.002218

Table 3. Comparison of Proposed Scheme with (Mukundan & Awasthi, 2015) at different space points with $h = 0.01$, $T = 2.3$ and $\nu = 0.1$.

	(Mukundan & Awasthi, 2015) $\Delta t = 0.01$ (230 time iteration)		Proposed Scheme $\Delta t = 0.023$ (100 time iteration)	
x	BDF-2	BDF-3	BDF-2	Exact Solution
0.1	0.02234	0.02253	0.02214	0.02214
0.2	0.04319	0.04357	0.04280	0.04280
0.3	0.06100	0.06155	0.06043	0.06043
0.4	0.07415	0.07485	0.07344	0.07344
0.5	0.08104	0.08182	0.08023	0.08023
0.6	0.08023	0.08104	0.07940	0.07940
0.7	0.07087	0.07161	0.07010	0.07011
0.8	0.05311	0.05368	0.05252	0.05252
0.9	0.02850	0.02881	0.02817	0.02817

Table 4. Comparison of Proposed Scheme with (Mukundan & Awasthi, 2015) at different space points and time with $\nu = 0.1$.

		(Mukundan & Awasthi, 2015) $\Delta t = 0.01$, h = 0.0125		Proposed Scheme $\Delta t = 0.02$, h = 0.01	
x	T	BDF-2	BDF-3	BDF-2	Exact Solution
0.25	2.4	4.756 E-002	4.755 E-002	4.755 E-002	4.755 E-002
	2.6	3.956 E-002	3.955 E-002	3.956 E-002	3.955 E-002
	3.0	2.721 E-002	2.720 E-002	2.720 E-002	2.720 E-002
0.5	2.4	7.270 E-002	7.268 E-002	7.269 E-002	7.269 E-002
	2.6	5.968 E-002	5.966 E-002	5.967 E-002	5.967 E-002
	3.0	4.021 E-002	4.020 E-002	4.021 E-002	4.021 E-002
0.75	2.4	5.594 E-002	5.593 E-002	5.593 E-002	5.593 E-002
	2.6	4.522 E-002	4.520 E-002	4.521 E-002	4.521 E-002
	3.0	2.978 E-002	2.977 E-002	2.977 E-002	2.977 E-002

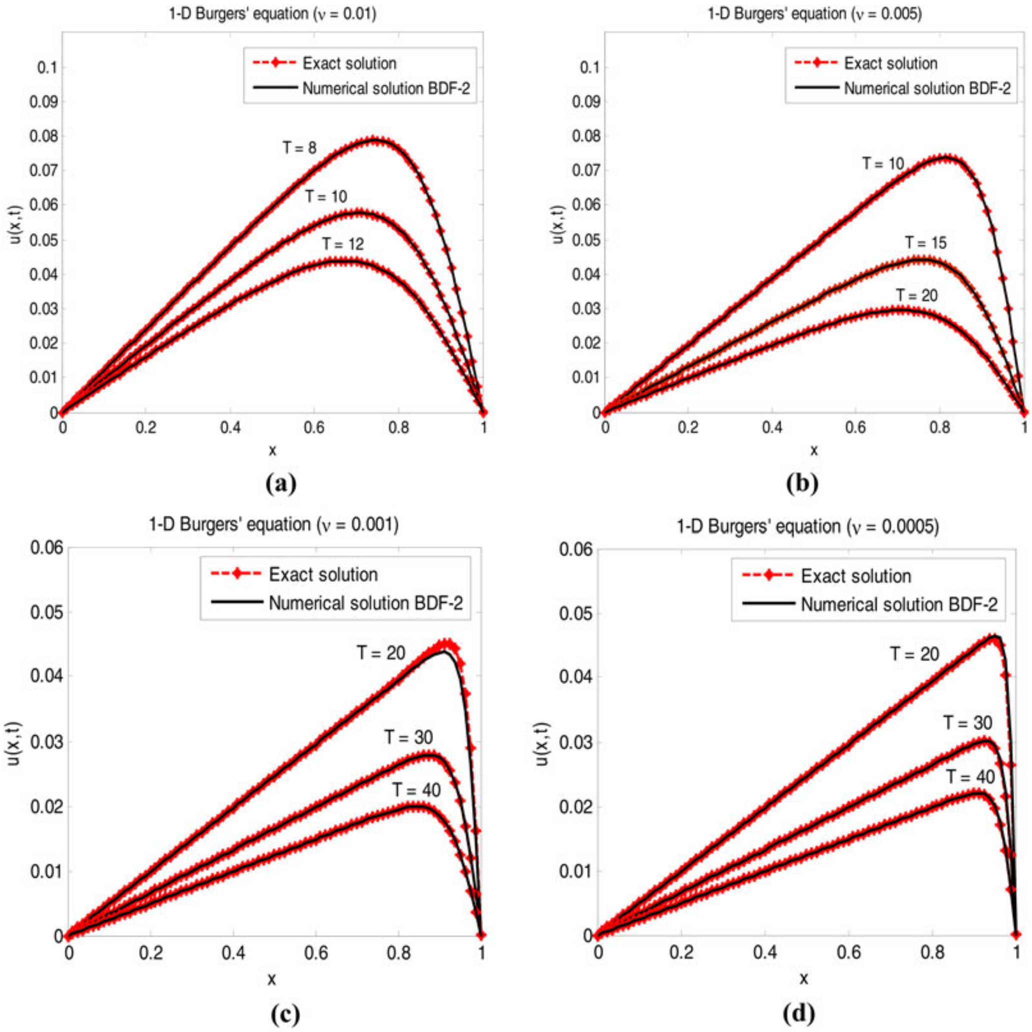

Figure 1. Comparison between the results obtained by Proposed Scheme and exact solution at different T, $h = 0.0125$ for (a) $\nu = 0.01$, $\Delta t = 0.025$ (b) $\nu = 0.005$, $\Delta t = 0.025$ (c) $\nu = 0.001$, $\Delta t = 0.01$ (d) $\nu = 0.0005$, $\Delta t = 0.01$.

Table 5. Comparison of Proposed Scheme with exact and other numerical methods at different space points and time with $\nu = 0.01$.

x	T	(Jiwari, 2012) $\Delta t = 0.001$	(Jiwari, 2015) $\Delta t = 0.001$	(Nojavan, Abbasbandy, & Mohammadi, 2018) N = 90	Proposed Scheme $\Delta t = 0.005$, N = 80	Exact
0.25	1.0	0.18815	0.18820	0.18819	0.18819	0.18819
	3.0	0.07510	0.07511	0.07511	0.07511	0.07511
0.5	1.0	0.37436	0.37443	0.37443	0.37441	0.37442
	3.0	0.15017	0.15019	0.15018	0.15018	0.15018
0.75	1.0	0.55608	0.55606	0.55607	0.55607	0.55605
	3.0	0.22502	0.22486	0.22484	0.22484	0.22481

$$\omega_{i,j} = -\frac{2}{3}\Delta t\left[\frac{\nu}{(\Delta y)^2} + \frac{\left(2u_{i,j}^{n} - u_{i,j}^{n-1}\right)}{2(\Delta y)}\right]$$

$$\left. f_{i,j} = \frac{4}{3}u_{i,j}^{n} - \frac{1}{3}u_{i,j}^{n-1} \right\}$$

Where Δx and Δy are constant spacing steps in x and y directions, respectively. Applying Eq. (23) at each point (i, j) results in a system of tri-diagonal matrix equation $Bu = F$ that can be solved efficiently by Thomas algorithm.

3. Error analysis

The error analysis of the discretized numerical scheme in Eq. (12) is obtained by using the Taylor Series Expansion. Eq. (12) can be written as:

$$u_i^{n+1} = \frac{4}{3}u_i^n - \frac{1}{3}u_i^{n-1} + \frac{2}{3}\Delta t \left[(\xi - \Omega_i)u_{i+1}^{n+1} - 2\xi u_i^{n+1} + (\xi + \Omega_i)u_{i-1}^{n+1}\right] \tag{24}$$

where,

$$\xi = \frac{\nu}{h^2} \text{ and } \Omega_i = \frac{2u_i^n - u_i^{n-1}}{2h}$$

Eq. (24) can be expressed as:

Table 6. Comparison of Proposed Scheme with (Mukundan & Awasthi, 2015) in order to get L_2 and L_∞ with different $\nu = 1,\ 0.1$ and $h = 0.0125$.

			(Mukundan & Awasthi, 2015)		Proposed Scheme
			BDF-2	BDF-3	BDF-2
V	*T*	Error	$\Delta t = 0.001$		$\Delta t = 0.002$
1	1	L_2	4.6043 E-08	4.8173 E-09	1.1488 E-08
		L_∞	6.5114 E-08	6.8127 E-09	1.6246 E-08
	2	L_2	5.0898 E-012	6.9285 E-013	6.5589 E-013
		L_∞	7.1885 E-012	9.7552 E-013	9.2756 E-013
			$\Delta t = 0.01$		$\Delta t = 0.025$
0.1	3	L_2	7.1682 E-006	1.5263 E-006	1.2204 E-006
		L_∞	1.0225 E-005	2.1892 E-006	1.9313 E-006
	3.5	L_2	5.6070 E-006	7.0422 E-007	1.9764 E-007
		L_∞	7.9505 E-006	1.0085 E-006	2.8145 E-007

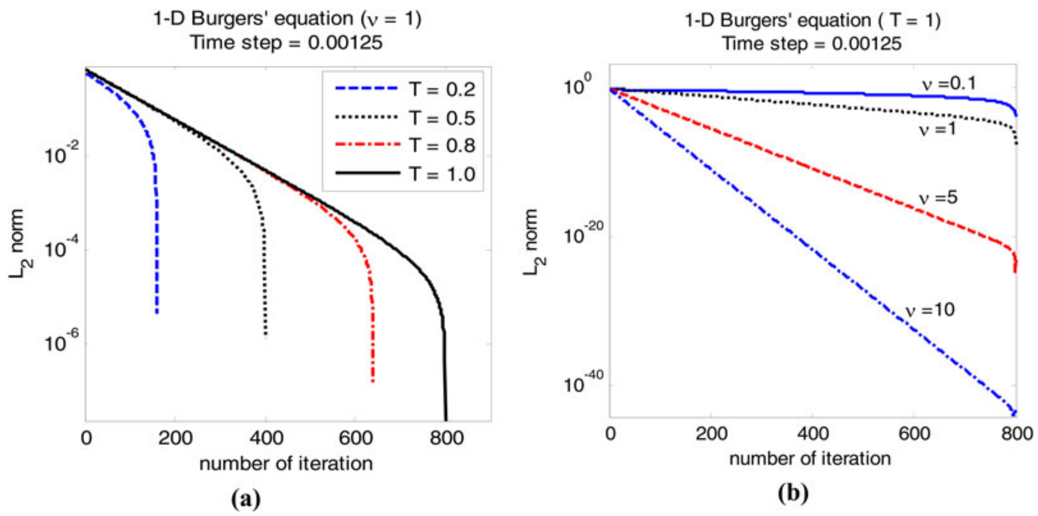

Figure 2. L_2- norm of Proposed Scheme for $\Delta t = 0.00125$ at (a) different T and fixed $\nu = 1$ (b) different ν and fixed $T = 1$.

$$u(x_i,\ t_{n+1}) = \frac{4}{3}u(x_i,\ t_n) - \frac{1}{3}u(x_i,\ t_{n-1}) + \frac{2}{3}\Delta t \left[(\xi - \Omega_i)u(x_{i+1},\ t_{n+1}) - 2\xi u(x_i,\ t_{n+1}) + (\xi + \Omega_i)u(x_{i-1},\ t_{n+1})\right] \tag{25}$$

Local Truncation Error (LTE) is obtained by:

$$LTE = u(x_i,\ t_{n+1}) - \frac{4}{3}u(x_i,\ t_n) + \frac{1}{3}u(x_i,\ t_{n-1}) - \frac{2}{3}\Delta t\ \left[(\xi - \Omega_i)u(x_{i+1},\ t_{n+1}) - 2\xi u(x_i,\ t_{n+1}) + (\xi + \Omega_i)u(x_{i-1},\ t_{n+1})\right] \tag{26}$$

Apply Taylor series expansion to Eq. (26) and simplify to get:

$$\begin{aligned}LTE = {} & \frac{2}{3}\Delta t\ u_t + \frac{2}{3}(\Delta t)^2\ u_{tt} + \frac{(\Delta t)^3}{9}\ u_{ttt} - \frac{2}{3}\Delta t \\ & \Big[(-2\Omega_i)h\ u_x + \frac{1}{2!}\left[(2\xi)h^2\ u_{xx} - (4\Omega_i)h\Delta t\ u_{xt}\right] \\ & + \frac{1}{3!}\left[(-2\Omega_i)h^3\ u_{xxx} + (6\xi h^2)\Delta t\ u_{xxt} - (6\Omega_i)h(\Delta t)^2\ u_{xtt}\right] + \frac{1}{4!}\left[(2\xi)h^4\ u_{xxxx} - (8\Omega_i)h(\Delta t)^3\ u_{xttt} + (12\xi h^2)(\Delta t)^2\ u_{xxtt} - (8\Omega_i)h^3\Delta t\ u_{xxxt} + (2\xi)(\Delta t)^4\ u_{tttt}\right]\Big] + \dots \\ = {} & \frac{2}{3}\ \Delta t\ \left[\ u_t - \nu\ u_{xx} + \left(2u_i^n - u_i^{n-1}\right)\ u_x\right] \\ & + \frac{2}{3}(\Delta t)^2\left[\ u_{tt} - \nu\ u_{xxt} + \left(2u_i^n - u_i^{n-1}\right)\ u_{xt}\right] \\ & + \frac{(\Delta t)^3}{3}\left[\frac{1}{3}\ u_{ttt} - \nu\ u_{xxtt} + \left(2u_i^n - u_i^{n-1}\right)\ u_{xtt}\right] \\ & + \left(-\frac{2}{3}\Delta t\right)\left[\frac{\nu h^2}{12}\ u_{xxxx} - \frac{h^2}{6}\left(2u_i^n - u_i^{n-1}\right)\ u_{xxx}\right] + \dots \\ = {} & (\Delta t)^3\left[\frac{1}{9}\ u_{ttt} - \frac{\nu}{3}\ u_{xxtt} + \frac{1}{3}\left(2u_i^n - u_i^{n-1}\right)\ u_{xtt}\right] \\ & + \Delta t\ h^2\left[\frac{-\nu}{18}\ u_{xxxx} + \frac{\left(2u_i^n - u_i^{n-1}\right)}{9}\ u_{xxx}\right] + \dots \\ = {} & O\left[(\Delta t)^3 + \Delta t\ h^2\right]\end{aligned}$$

Truncation Error (TE) is computed by:

$$TE = (\Delta t)^{-1}(LTE) = O\left[(\Delta t)^2 + h^2\right]$$

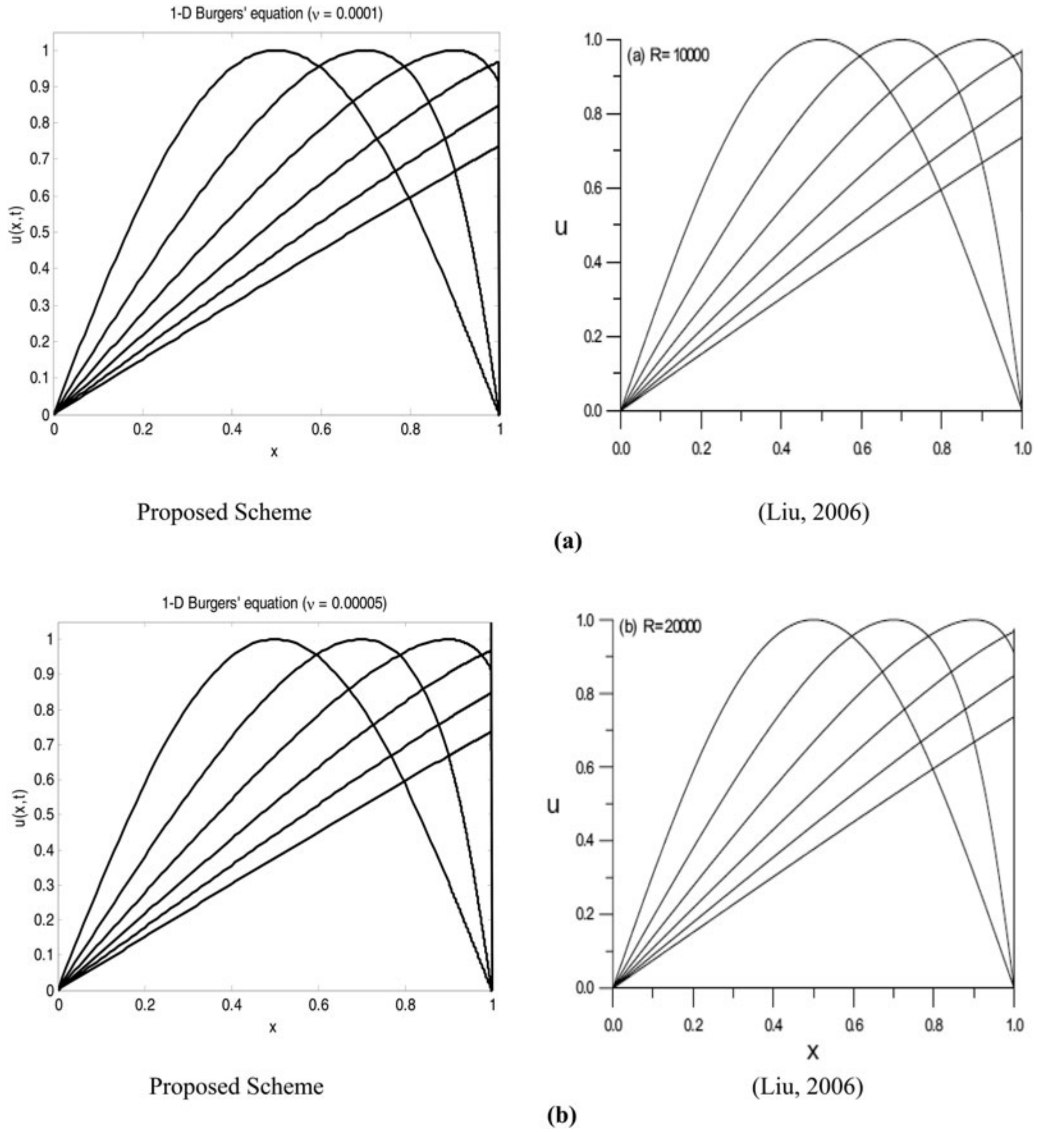

Figure 3. The numerical solution of the Proposed Scheme and the comparison with (Liu, 2006) at $h = 0.01$,.

Table 7. Comparison of proposed scheme with (Mukundan & Awasthi, 2015) at different space points with T = 1 and $\nu = 1$.

	(Mukundan & Awasthi, 2015) $\Delta t = 0.001$, h = 0.0125		Proposed Scheme $\Delta t = 0.00125$, h = 0.01	Exact Solution
x	BDF-2	BDF-3	BDF-2	
0.1	1.646 E-005	1.644 E-005	1.644 E-005	1.644 E-005
0.2	3.131 E-005	3.128 E-005	3.128 E-005	3.127 E-005
0.3	4.310 E-005	4.305 E-005	4.305 E-005	4.304 E-005
0.4	5.066 E-005	5.061 E-005	5.061 E-005	5.060 E-005
0.5	5.327 E-005	5.321 E-005	5.321 E-005	5.320 E-005
0.6	5.066 E-005	5.061 E-005	5.061 E-005	5.060 E-005
0.7	4.310 E-005	4.305 E-005	4.305 E-005	4.304 E-005
0.8	3.131 E-005	3.128 E-005	3.128 E-005	3.127 E-005
0.9	1.646 E-005	1.644 E-005	1.644 E-005	1.644 E-005

So, the errors are proportional to the square of time step and square of space step.

4. The numerical results

Four test problems of Burgers' Equation are solved to demonstrate the accuracy and efficiency of the proposed numerical schemes; and the numerical results are compared with exact solution and the numerical results which are listed in previous references at different nodal points and at different final time T for different values of kinematic viscosity ν.

Since the exact solution is given, the L_2 and L_∞ error norms are computed after each time step by using the following definitions:

$$L_2 := \left|\left|u_{exact} - u_{computed}\right|\right|_2 = \sqrt{\left(\sum_{j=1}^{n} \frac{\left|u_j^{exact} - u_j^{computed} 2\right)}{N}\right.}$$

Table 8. Comparison of Proposed Scheme with (Mukundan & Awasthi, 2015) at different space points and time with $\nu = 0.1$.

		(Mukundan & Awasthi, 2015) $\Delta t = 0.01$, $h = 0.0125$		Proposed Scheme $\Delta t = 0.025$, $h = 0.01$	
x	T	BDF-2	BDF-3	BDF-2	Exact Solution
0.25	2.2	5.814 E-002	5.813 E-002	5.814 E-002	5.814 E-002
	2.4	4.850 E-002	4.849 E-002	4.849 E-002	4.849 E-002
	2.6	4.035 E-002	4.034 E-002	4.035 E-002	4.035 E-002
0.5	2.2	9.046 E-002	9.044 E-002	9.045 E-002	9.045 E-002
	2.4	7.426 E-002	7.424 E-002	7.424 E-002	7.424 E-002
	2.6	6.096 E-002	6.094 E-002	6.094 E-002	6.094 E-002
0.75	2.2	7.110 E-002	7.108 E-002	7.108 E-002	7.109 E-002
	2.4	5.724 E-002	5.723 E-002	5.723 E-002	5.723 E-002
	2.6	4.625 E-002	4.624 E-002	4.624 E-002	4.624 E-002

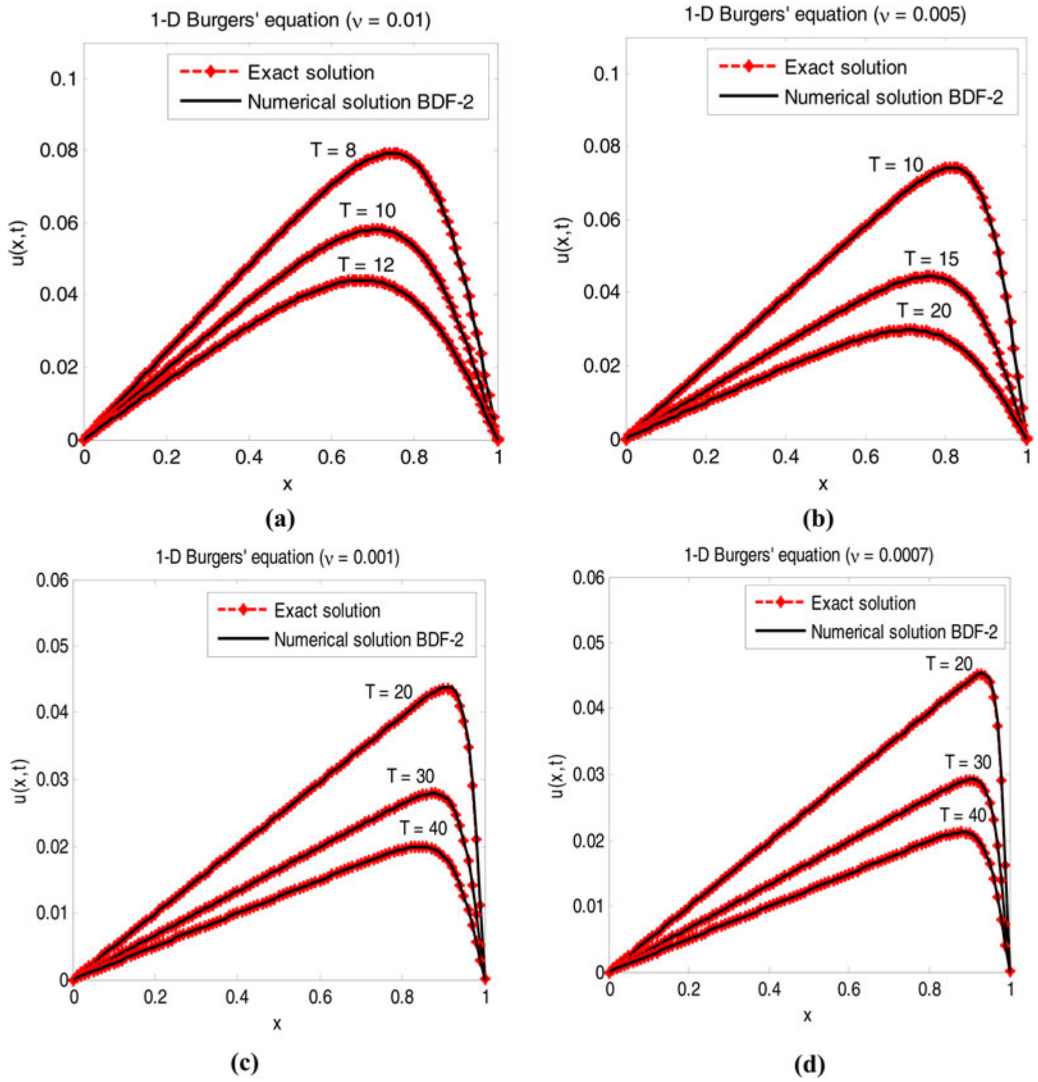

Figure 4. The comparison between the results obtained by the Proposed Scheme and exact solution at different T, $h = 0.01$ for (a) $\nu = 0.025$, $\Delta t = 0.025$ (b) $\nu = 0.005$, $\Delta t = 0.025$ (c) $\nu = 0.001$, $\Delta t = 0.01$ (d) $\nu = 0.0007$, $\Delta t = 0.01$.

$$L_\infty := \left|\left|u_{exact} - u_{computed}\right|\right|_\infty = max_j \left|u_j^{exact} - u_j^{computed}\right.$$

Where u_{exact} and $u_{computed}$ are the exact and computed solutions at each point, respectively.

For the cases with high Reynolds number (Re), the results are obtained by using the numerical Proposed Scheme that is applied to the non- uniform spacing scheme. In all the examples the initial and boundary conditions are taken from the exact solutions. The computational work has been done by using Matlab software.

Example 1. Solve Burgers' Eqs. (1–3a) with

Initial condition:

$$u(x, 0) = f(x) = \sin(\pi x), \quad 0 \leq x \leq 1$$

and the boundary conditions

$$u(0, t) = u(1, t) = 0, \; 0 \leq t \leq T$$

whose exact solution is given by Cole Hopf as in

Table 9. Comparison of proposed scheme with exact and other numerical methods at different space points and time with $\nu = 0.01$.

x	T	(Jiwari, 2012)$\Delta t = 0.001$	(Mukundan & Awasthi, 2017) $\Delta t = 0.001$	Proposed Scheme $\Delta t = 0.005$, N = 80	Exact
0.25	0.4	0.36217	0.36244	0.36226	0.36226
	1.0	0.19465	0.19477	0.19470	0.19469
	3.0	0.07613	0.07615	0.07613	0.07613
0.5	0.4	0.68357	0.68398	0.68371	0.68368
	1.0	0.38561	0.38586	0.38568	0.38568
	3.0	0.15217	0.15222	0.15218	0.15218
0.75	0.4	0.92050	0.92079	0.92067	0.92050
	1.0	0.56924	0.56959	0.56934	0.56932
	3.0	0.22774	0.22784	0.22777	0.22774

Table 10. Comparison of Proposed Scheme with (Mukundan & Awasthi, 2015) to get L_2 and L_∞ with $T = 5$, 5.2 and $\nu = 0.05$.

			(Mukundan & Awasthi, 2015) $\Delta t = 0.01$, $h = 0.0125$		Proposed Method $\Delta t = 0.025$, $h = 0.01$
ν	Error	T	BDF-2	BDF-3	BDF-2
0.05	L_2	5	5.3991 E-05	4.4162 E-05	3.4021 E-07
	L_∞		8.9093 E-06	6.9974 E-06	5.4531E-07
	L_2	5.2	5.2835 E-05	3.9302 E-05	2.9785E-07
	L_∞		8.6442 E-06	6.2250 E-06	5.2886E-07

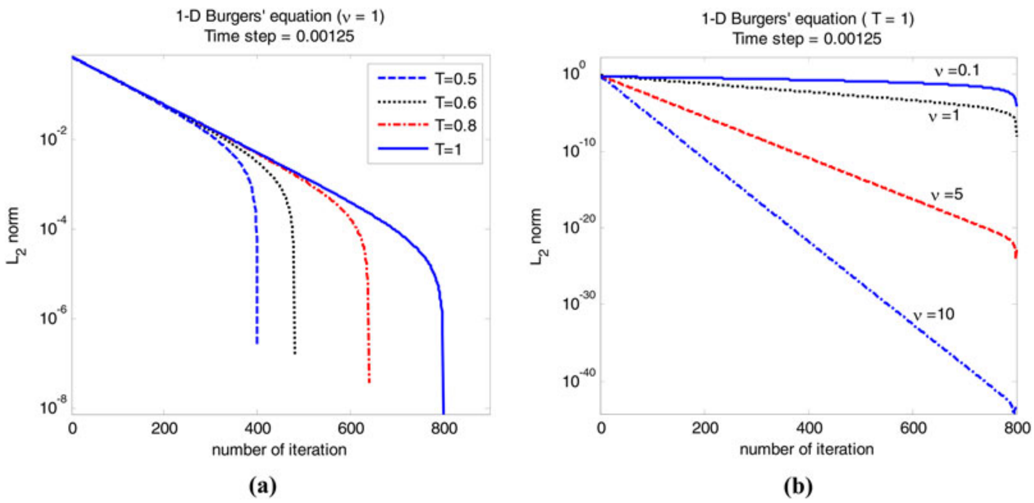

Figure 5. L_2 norm of Proposed Scheme for $\Delta t = 0.00125$ at (a) different T and fixed $\nu = 1$ (b) different ν and fixed $T = 1$.

(Mukundan & Awasthi, 2015):

$$u(x,t) = 2\nu\pi \frac{\sum_{n=1}^{\infty} c_n \exp(-n^2\pi^2\nu t)\ n\ \sin(n\pi x)}{c_0 + \sum_{n=1}^{\infty} c_n \exp(-n^2\pi^2\nu t)\ \cos(n\pi x)} \tag{27}$$

where the Fourier coefficients c_0 and c_n are:

$$c_0 = \int_0^1 \exp\left\{-\frac{1}{2\nu\pi}[1-\cos(\pi x)]\right\} dx$$

$$c_n = 2\int_0^1 \exp\left\{-\frac{1}{2\nu\pi}[1-\cos(\pi x)]\right\} \cos(n\pi x)\ dx$$

In this problem, the numerical solutions are presented for $\nu = 10,\ 1, 0.1$ to get the solutions at final time $T = 0.1, 0.5,\ 2.3$, respectively in Tables 1–3. Results show that the Proposed Scheme that used BDF-2 is more accurate than that used in (Mukundan & Awasthi, 2015). Mukundan and Awasthiin (Mukundan & Awasthi, 2015) needed to solve more iterations and apply BDF-3 in order to get the accuracy that obtained in the Proposed Scheme. In Table 4, results are also shown at different times and a good agreement is still found with the earlier work.

Figure 1 shows the comparison between the numerical results obtained by Proposed Scheme and exact solution for different values of T, h =0.0125 and Δt =0.025, 0.01 at $\nu = 0.01,\ 0.005,\ 0.001,\ 0.0005$. It is clear that Proposed Scheme is accurate even for small values of ν. In Table 5, results of the proposed scheme have been compared with other schemes; and excellent agreement is still achieved.

Table 6 shows the comparison of Proposed Scheme in term of L_2 and L_∞ errors for different values of time at $\nu = 1,\ 0.1$ with the existing method in (Mukundan & Awasthi, 2015). The L_2 error in Figure 2 shows the effect of changing time T at fixed ν on the solution and *vice versa*. From Table 6 and

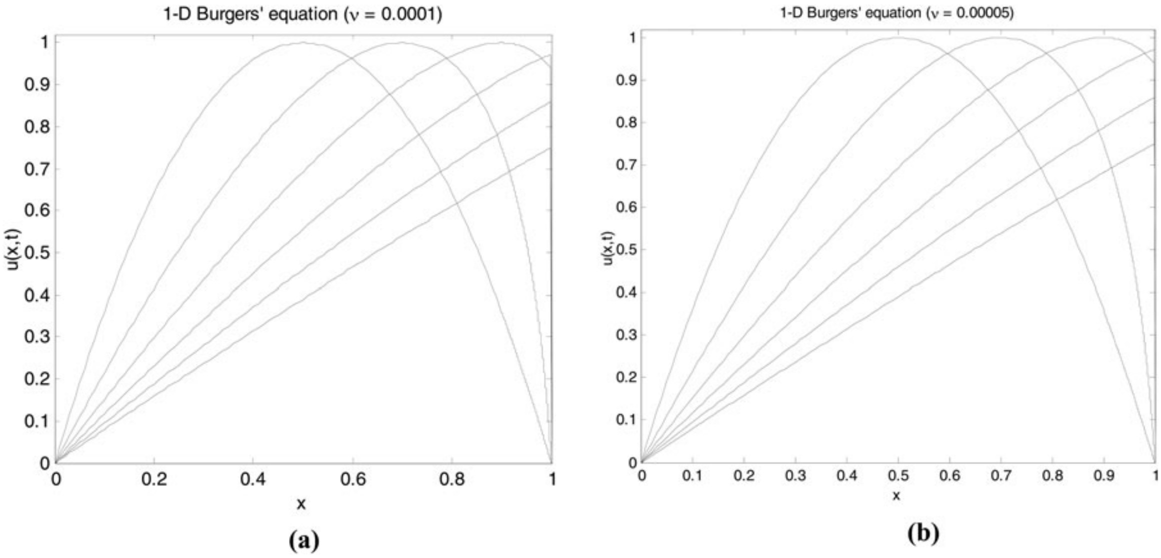

Figure 6. The numerical solution of Proposed Scheme at $h = 0.01$, $\Delta t = 0.0001$ and different time (six curves at $T = 0$, 0.2, 0.4, 0.6, 0.8, 1.0 from left to right) for (a) $Re = 10000$ (b) $Re = 20000$.

Table 11. L_2 and L_∞ errors norm with different T, Re at $\Delta t = 0.001$ and $h = 0.025$.

	$Re = 20$		$Re = 100$	
T	L_∞	L_2	L_∞	L_2
0.01	1.1 E-03	3.9 E-04	8.3 E-04	5.6 E-04
0.05	5.0 E-03	1.8 E-03	4.1 E-03	2.8 E-03
0.1	8.4 E-03	3.2 E-03	8.3 E-03	5.6 E-03

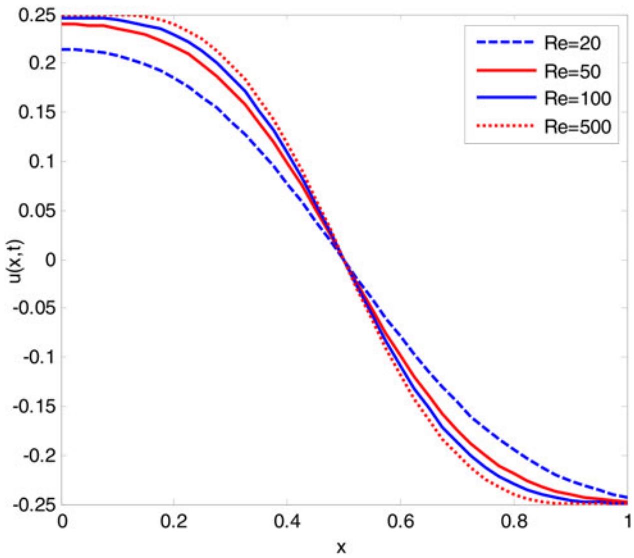

Figure 7. The numerical solution of Proposed Scheme at $h = 0.025$, $\Delta t = 0.005$ and $T = 0.5$ for different Re.

Figure 2, it is observed that the accuracy and efficiency of the Proposed Scheme have been achieved.

As viscosity gets smaller and smaller, the graphs demonstrate a very sharp front near the left boundary and the steepness of the solution that increases sharply near the right boundary. This steepness is controlled by using the non-uniform scheme with $l_2 = 0.1$ and $\varepsilon = 1/50$. Figure 3 concludes that the present results at different values of times are in good agreement with the results in (Liu, 2006) at $Re = 10000$, 20000.

$\Delta t = 0.0001$ and different time (six curves at $T = 0.1$, 0.2, 0.4, 0.6, 0.8, 1.0 from left to right) for (a) $Re = 10000$ (b) $Re = 20000$

Example 2. In this example Burgers' Eqs. (1–3a) are solved with

Initial condition:

$$u(x,0) = 4x\,(1 - x), 0 \le x \le 1$$

and the boundary conditions

$$u(0,t) = u(1,t) = 0, 0 \le t \le T$$

Table 12. L2 and L_∞ errors norm for different *Re* at $\Delta t = 0.005$.

		$T = 0.05$		$T = 0.25$	
Grid size	*Re*	L_∞	L_2	L_∞	L_2
5×5	1	4.3309 E-006	2.2191 E-006	7.1482 E-006	3.5268 E-006
10×10		1.3062 E-006	6.9661 E-007	1.9616 E-006	9.9315 E-007
20×20		4.7204 E-007	2.5389 E-007	5.0561 E-007	2.6815 E-007
		$T = 0.05$		$T = 0.25$	
L_∞	L_2	L_∞	L_2		
10×10	10	5.5763 E-004	8.0068 E-005	1.8010 E-003	3.9226 E-004
20×20		1.7575 E-004	2.7495 E-005	4.9601 E-004	1.1061 E-004
30×30		9.3799 E-005	1.6417 E-005	2.3733 E-004	5.2981 E-005

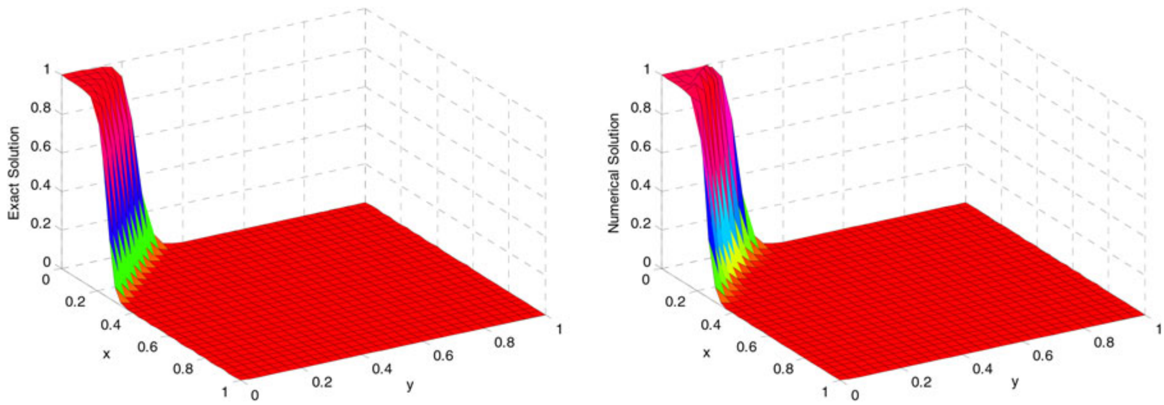

Figure 8. The exact and numerical solution of Example 3 for $Re = 100$, grid size $= 30 \times 30$, $\Delta t = 0.0005$ and $T = 0.25$.

where exact solution is given by Eq. (27) and the Fourier coefficients c_0 and c_n are, as in (Mukundan & Awasthi, 2015):

$$c_0 = \int_0^1 \exp\left\{-\frac{1}{3\nu}\left[x^2(3-2x)\right]\right\}\, dx$$

$$c_n = 2\int_0^1 \exp\left\{-\frac{1}{3\nu}\left[x^2(3-2x)\right]\right\} cos(n\pi x)\, dx$$

In this problem, the numerical solutions are presented for $\nu = 1$ in order to get the solutions at final time $T = 1$ in Table 7. Results show that the Proposed Scheme that used BDF-2 is more accurate than used in (Mukundan & Awasthi, 2015). Mukundan and Awasthi in (Mukundan & Awasthi, 2015) needed to solve more iterations and apply BDF-3 to get the accuracy that obtained in the Proposed Scheme. In Table 8, results are also shown at different times and a good agreement is still found with the earlier work.

Figure 4 shows the comparison between the numerical results obtained by the Proposed Scheme and exact solution for different values of T, $h = 0.01$ and $\Delta t = 0.025, 0.01$at $\nu = 0.01, 0.005, 0.001, 0.0007$.It is clear that the Proposed Scheme is efficient to obtain good agreement with the exact solution even for large values of Reynolds number *Re*. In Table 9, it is observed that the results of the proposed scheme are still achieving a good agreement compared with other schemes.

Table 10 shows the comparison of the computed results in term of L_2 and L_∞ errors for different values of time at $\nu = 0.05$ with the existing method in (Mukundan & Awasthi, 2015). L_2 error in Figure 5 shows the effect of changing time T at fixed ν on the solution and *vice versa*. From Table 10 and Figure 5, it is observed that the accuracy and efficiency of the Proposed Scheme have been achieved.

As viscosity gets smaller and smaller, the graphs demonstrate a very sharp front near the left boundary and the steepness of the solution increases sharply near the right boundary. This steepness is controlled by using the non-uniform scheme with $l_2 = 0.1$ and $\varepsilon = 1/50$. Figure 6 concludes that the present results at different values of times are in good agreement with the results in (Liu, 2006) at $Re = 10000$, 20000.

Example 3. Solve one-dimensional Burgers' Eqs. (1-3b): with Initial condition:

$$u(x, 0) = \frac{1}{4}\ \cos(\pi x),\ \ 0\ \le\ x\ \le\ 1,$$

and the boundary conditions:

$$u_x(0, t) = 0,\ \ u(1, t) = -\frac{1}{4}\ e^{-\nu t}\ \ t > 0,$$

for which the exact solution is given as in (Pugh, 1995):

$$u(x, t) = \frac{1}{4}\ e^{-\nu t}\cos(\pi x),\ \ 0\ \le\ x\ \le\ 1.$$

Error norms L_2 and L_∞ are computed for $Re = 20$, 100 at different time levels and listed in Table 11.

In Figure 7, the numerical solutions are presented to show the effect of changing *Re* at fixed *T*. It is observed from Table 11 and Figure 7 that the Proposed Scheme is efficient for solving the mixed boundary condition Burgers' equation.

Example 4. Solve two-dimensional Burgers' Eqs. (4–6) over a square domain $\Omega: \ [0,1]\times[0,1]$ with

Initial condition:

$$u(x,y,0)=\frac{1}{1+e^{\frac{Re(x+y)}{2}}},\quad (x,y)\in\Omega$$

and the boundary conditions:

$$u(0,y,t)=\frac{1}{1+e^{\frac{Re(y-t)}{2}}},\quad u(1,y,t)=\frac{1}{1+e^{\frac{Re(1+y-t)}{2}}},\quad y\in[0,1],\ \ t>0,$$

$$u(x,0,t)=\frac{1}{1+e^{\frac{Re(x-t)}{2}}},\quad u(x,1,t)=\frac{1}{1+e^{\frac{Re(1+x-t)}{2}}},\quad x\in[0,1],\ \ t>0,$$

for which the exact solution is given as in (Liu, Pope, & Sepehrnoori, 1995):

$$u(x,y,t)=\frac{1}{1+e^{\frac{Re(x+y-t)}{2}}},\quad (x,y)\in\Omega,\ \ t\geq 0.$$

The computed solutions of Eqs. (4–6) have been obtained with the above initial and boundary conditions by using the Proposed Scheme for different grid size and Reynolds number (*Re*). In Table 12 the error norms L_2 and L_∞ with different grid sizes and time at $Re=1,10$ are listed. It is observed from Table 12 that the changing of the grid size is efficient to get the suitable accuracy of the Proposed Scheme. The comparison between the exact solution and numerical solution for $Re=100$, grid size $=30\times30$, $\Delta t=0.0005$ and $T=0.25$ is shown in Figure 8. Figure 8 shows the solution behaviour of the Proposed Scheme which is similar to the exact solution.

5. Conclusions

In this study the author presents efficient numerical algorithms based on finite difference method in order to solve one and two dimensions unsteady Burgers' equation without transformation of the equation. The proposed schemes are examined on four examples of Burgers' equation with different values of ν and all the numerical results that are obtained demonstrating the efficiency and accuracy of the considered schemes and approaching the exact solutions. It is evident that the proposed schemes using backward differentiation formula of order two produce accurate results compared with the exact solution and better than other schemes available in the literature. Even for very high Reynolds number the scheme is applicable under a reasonable grid spacing length and time step size compared with (Liu, 2006). Also, this paper proposes a scheme which has succeeded in solving 1) D. Burgers' equation with mixed boundary conditions, and obtain the accurate solutions of 2) D. Burgers' equation with different parameters. The proposed schemes can be easily implemented and have the possibility of expanding to solve Coupled Burgers' equations in two-dimensional and other non-linear problems in fluid dynamics.

Acknowledgements

The author thanks the reviewers for their valuable suggestions and comments.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Abdou, M. A., & Soliman, A. A. (2005). Variational iteration method for solving Burgers and coupled Burgers' equations. *Journal of Computational and Applied Mathematics*, *181*(2), 245–251.

Aksan, E. N. (2006). Quadratic B-spline finite element method for numerical solution of the Burgers' equation. *Applied Mathematics and Computation*, *174*(2), 884–896.

Ali, A. H. A., Gardner, G. A., & Gardner, L. R. T. (1992). A collocation solution for Burgers' equation using cubic B-spline finite elements. *Computer Methods in Applied Mechanics Engineering*, *100*(3), 325–337.

Arafa, A. R. A. (1996). Analytical numerical method for solving nonlinear partial differential equations. *Applied Mathematics Letter*, *9*(4), 115–122.

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

Asaithambi, A. (2010). Numerical solution of the Burgers' equation by automatic differentiation. *Applied Mathematics and Computation*, *216*(9), 2700–2708.

Atkinson, K. E., Han, W., & Stewart, D. (1999). *Numerical solutions of ordinary differential equations*. Hoboken, NJ: John Wiley and Sons, Inc, pp. 140–141.

Bateman, H. (1915). Some recent researches on the motion of fluids. *Monthly Weather Review*, *43*(4), 163–170.

Biazar, J., & Aminikhah, H. (2009). Exact and numerical solutions for non-linear burgers' equation by VIM. *Mathematical and Computer Modelling*, *49*(7–8), 1394–1400.

Burgers, J. M. (1939). Mathematical examples illustrating relations occurring in the theory of turbulent fluid motion. *Transaction of Royal Netherlands Academy of Science*, *17*(2), 1–53. Retrieved from http://www.dwc.knaw.nl/DL/publications/PU00011461.pdf.

Burgers, J. M. (1948). *A mathematical model illustrating the theory of turbulence*. Delft, Holland: Technical University, vol. 1, pp. 171–199.

Ciment, M., Leventhal, S. H., & Weinberg, B. C. (1978). The operator compact implicit method for parabolic equations. *Journal of Computational Physics*, *28*(2), 135–166.

Cole, J. D. (1951). On a quasi-linear parabolic equation in aerodynamics. *Quarterly of Applied Mathematics*, *9*(3), 225–236.

Dehghan, M., Hamidi, A., & Shakourifar, M. (2007). The solution of coupled Burgers' equations using Adomian: pade technique. *Applied Mathematics and Computation*, *189*(2), 1034–1047.

Dogan, A. (2004). A Galerkin finite element approach to Burgers' equation. *Applied Mathematics and Computation*, *157*(2), 331–346.

Ganaie, I. A., & Kukreja, V. K. (2014). Numerical solution of Burgers' equation by cubic Hermite collocation method. *Applied Mathematics and Computation*, *237*, 571–581.

Hassanien, I. A., Salama, A. A., & Hosham, H. A. (2005). Fourth-order finite difference method for solving Burgers' equation. *Applied Mathematics and Computation*, *170*(2), 781–800.

Heinrichs, W. (2007). An adaptive spectral least-squares scheme for the Burgers' equation. *Numerical Algorithms*, *44*(1), 1–10.

Hopf, E. (1950). The partial differential equation $Ut + UUx = \mu Uxx$. *Communications on Pure and Applied Mathematics*, *3*(3), 201–230.

Iskandar, L., & Mohsen, A. (1992). Some numerical experiments on the splitting of Burgers' equation. *Numerical Methods for Partial Differential Equations*, *8*(3), 267–276.

Jiwari, R. (2012). A Haar wavelet quasilinearization approach for numerical simulation of Burgers' equation. *Computer Physics Communications*, *183*(11), 2413–2423.

Jiwari, R. (2015). A hybrid numerical scheme for the numerical solution of the Burgers' equation. *Computer Physics Communications*, *188*, 59–67.

Kadalbajoo, M. K., & Gupta, V. (2010). A brief survey on numerical methods for solving singularly perturbed problems. *Applied Mathematics and Computation*, *217*(8), 3641–3716.

Kay, D. A., Gresho, P. M., Griffiths, D. F., & Silvester, D. J. (2010). Adaptive time-stepping for incompressible flow part II: Navier-Stokes equations. *Siam Journal of Scientific Computing*, *32*(1), 111–128.

Khalifa, A. K., Noor, K. I., & Noor, M. A. (2011). Some numerical methods for solving Burgers' equation. *International Journal of the Physical Sciences*, *6*(7), 1702–1710.

Khan, M. (2014). A novel solution technique for two dimensional Burgers' equation. *Alexandria Engineering Journal*, *53*(2), 485–490.

Khater, A. H., Temsah, R. S., & Hassan, M. M. (2008). A Chebyshev spectral collocation method for solving Burgers' type equations. *Journal of Computational and Applied Mathematics*, *222*(2), 333–350.

Korkmaz, A., & Dağ, İ. (2011a). Shock wave simulations using sinc differential quadrature method. *International Journal for Computer–Aided Engineering and Software*, *28*(6), 654–674.

Korkmaz, A., & Dağ, I. (2011b). Polynomial based differential quadrature method for numerical solution of nonlinear Burgers' equation. *Journal of the Franklin Institute*, *348*(10), 2863–2875.

Kutulay, S., Bahadir, A. R., & Odes, A. (1999). Numerical solution of the one-dimensional Burgers' equation: Explicit and exact-explicit finite difference methods. *Journal of Computational and Applied Mathematics*, *103*, 251–261.

Liu, C. S. (2009). A fictitious time integration method for the Burgers' equation. *Computers, Materials, & Continua*, *9*(3), 229–252. Retrieved from http://www.techscience.com/doi/10.3970/cmc.2009.009.229.html.

Liu, S. (2006). A group preserving scheme for Burgers equation with very large Reynolds number. *Computer Modeling in Engineering and Sciences*, *12*(3), 197–211. Retrieved from http://www.techscience.com/doi/10.3970/cmes.2006.012.197.pdf.

Liu, J., Pope, G. A., & Sepehrnoori, K. (1995). A high-resolution finite-difference scheme for non-uniform grids. *Applied Mathematics and Modelling*, *19*(3), 162–172.

Maerschalck, B. D., & Gerritsma, M. I. (2005). The use of Chebyshev polynomials in the space-time least-squares spectral element method. *Numerical Algorithms*, *38*, 173–196.

Maerschalck, B. D., & Gerritsma, M. I. (2008). Least-squares spectral element method for non-linear differential equations. *Journal of Computational and Applied Mathematics*, *215*(2), 357–367.

Mittal, R. C., & Jain, R. K. (2012). Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method. *Applied Mathematics and Computation*, *218*(15), 7839–7855.

Mukundan, V., & Awasthi, A. (2015). Efficient numerical techniques for Burgers' equation. *Applied Mathematics and Computation*, *262*, 282–297.

Mukundan, V., & Awasthi, A. (2017). Numerical techniques for unsteady nonlinear burgers equation based on backward differentiation formulas. *Nonlinear Engineering*, *7*(3), 171–181.

Nojavan, H., Abbasbandy, S., & Mohammadi, M. (2018). Local variably scaled Newton basis functions collocation method for solving Burgers' equation. *Applied Mathematics and Computation*, *330*, 23–41.

Ozis, T., Aksan, E. N., & Ozdes, A. (2003). A finite element approach for solution of Burgers' equation. *Applied Mathematics and Computation*, *139*, 417–428.

Ozis, T., & Erdogan, U. (2009). An exponentially fitted method for solving Burgers' equation. *International Journal for Numerical Methods in Engineering*, *79*, 696–705.

Pugh, S. M. (1995). Finite element approximation of Burgers' equation [M.S. Thesis]. Virginia Polytechnic Institute and State University.

Radwan, S. F. (2005). Comparison of higher-order accurate schemes for solving the two-dimensional unsteady Burgers' equation. *Journal of Computational and Applied Mathematics*, *174*(2), 383–397.

Rashidi, M. M., Domairry, G., & Dinarvand, S. (2009). Approximate solutions for the Burgers' and regularized long wave equations by means of the homotopy analysis method. *Communications in Nonlinear Science and Numerical Simulation*, *14*(3), 708–717.

Rashidi, M. M., & Erfani, E. (2009). New analytic method for solving Burgers' and nonlinear heat transfer equations and comparison with HAM. *Computer Physics Communications*, *180*(9), 1539–1544.

Saka, B., & Dağ, İ. (2007). Quartic B-spline collocation method to the numerical solutions of the Burgers' equation. *Chaos, Solitons and Fractals*, *32*(3), 1125–1137.

Seadawy, A. R. (2015a). Nonlinear wave solutions of the three-dimensional Zakharov–Kuznetsov Burgers equation in dusty plasma. *Physica A, Statistical Mechanics and Its Applications*, *439*, 124–131.

Seadawy, A. R. (2015b). Approximation solutions of derivative nonlinear Schrdinger equation with computational applications by variational method. *The European Physical Journal Plus*, *130*, 182, 1–10.

Seadawy, A. R. (2017a). Ion acoustic solitary wave solutions of two-dimensional nonlinear Kadomtsev Petviashvili–Burgers equation in quantum plasma. *Mathematical Methods in the Applied Sciences*, *40*(5), 1598–1607.

Seadawy, A. R. (2017b). Solitary wave solutions of two-dimensional nonlinear Kadomtsev Petviashvili dynamic equation in dust-acoustic plasmas. *Pramana*, *89*, 49,1–11.

Seadawy, A. R. (2017c). Travelling-wave solutions of a weakly nonlinear two-dimensional higher-order Kadomtsev-Petviashvili dynamical equation for dispersive shallow-water waves. *The European Physical Journal Plus*, *132*, 29, 1–13.

Seydaoğlu, M., Erdoğan, U., & Öziş, T. (2016). Numerical Solution of Burgers' Equation with High Order Splitting Methods. *Journal of Computational and Applied Mathematics*, *291*, 410–421.

Sheikholeslami, M., & Zeeshan, A. (2017). Analysis of flow and heat transfer in water based nanofluid due to magnetic field in a porous enclosure with constant heat flux using CVFEM. *Computer Methods in Applied Mechanics and Engineering*, *320*, 68–81.

Sheikholeslami, M., & Zeeshan, A. (2018). Numerical simulation of Fe3O4-water nanofluid flow in a non-Darcy porous media. *International Journal of Numerical Methods for Heat & Fluid Flow*, *28*, 641–660.

Sheikholeslami, M., Zeeshan, A., & Majeed, A. (2018). Control volume based finite element simulation of magnetic nanofluid flow and heat transport in non-Darcy medium. *Journal of Molecular Liquids*, *268*, 354–364.

Tamsir, M., Srivastava, V., & Jiwari, R. (2016). An algorithm based on exponential modified cubic B-spline differential quadrature method for nonlinear Burgers' equation. *Applied Mathematics and Computation*, *290*, 111–124.

Xu, M., Wang, R. H., Zhang, J. H., & Fang, Q. (2011). A novel numerical scheme for solving Burgers' equation. *Applied Mathematics and Computation*, *217*(9), 4473–4482.

Zhang, D. S., Wei, G. W., Kouri, D. J., & Hoffman, D. K. (1997). Burgers' equation with high Reynolds number. *Physics Fluids*, *9*(6), 1853–1855. Retrieved from https://aip.scitation.org/doi/10.1063/1.869305.

Zhanlav, T., Chuluunbaatar, O., & Ulziibayar, V. (2015). Higher-order accurate numerical solution of unsteady Burgers' equation. *Applied Mathematics and Computation*, *250*, 701–707.