Available online at http://ijim.srbiau.ac.ir/
Int. J. Industrial Mathematics (ISSN 2008-5621)
Vol. 10, No. 1, 2018 Article ID IJIM-00957, 19 pages
Research Article

Comparison of the LBM With the Modified Local Crank-Nicolson Method Solution of Transient Two-Dimensional Non-Linear Burgers Eˆquation

AR. Haghighi *†, S. Pakrou ‡, AR. Dadvand §

Received Date: 2016-10-12 Revised Date: 2017-03-09 Accepted Date: 2017-04-12

Abstract

Burgers equation is a simplified form of the Navier-Stokes equation that represents the non-linear features of it. In this paper, the transient two-dimensional non-linear Burgers equation is solved using the Lattice Boltzmann Method (LBM). The results are compared with the Modified Local Crank-Nicolson method (MLCN) and exact solutions. The LBM has been emerged as a new numerical method for solving various physical problems. Compared with the traditional computational approaches, the LBM has many outstanding advantages, such as its algorithmic simplicity, parallel computation and easy handling of complicated boundary conditions. Two examples, distinguished by different initial conditions, are solved using the LBM and the MLCN method and the accuracy of these two methods at various Reynolds numbers are analyzed. Also, the effects of different numbers of particle velocities on the accuracy of the LBM are evaluated. The results show that at higher Reynolds numbers the accuracy of the LBM is higher than the MLCN method.

Keywords : Non-linear Burgers equation; Adomian method; The modified Local Crank-Nicolson method

1 Introduction

THe main problem concerned with computing the fluid dynamic problems arises from the inability to efficiently balance the non-linear convection terms and the diffusion term [1] in the Navier-Stokes equations. The nonlinear Burgers equation is a simplified form of the Navier-Stokes equations where the pressure and continuity terms have been omitted, and the remained equation is a mixture of convection and diffusion terms only. This equation has found a wide range of applications in engineering and physics. It is used as a generic model for turbulence, boundary layer flow, shock wave, gas dynamics, plasma dynamics, longitudinal elastic waves in an isotropic solid, acoustic attenuation in fog, continuum traffic simulation, shallow water waves, and the chemical reaction-diffusion model of Brusselator [2]-[11]. In addition, this equation is a simple model to investigate the fluid suspensions or colloids under the effect of gravity [12, 13]. There are several branches of science including chemistry, biology, mathematics, communication, solid-state physics, plasma

*Corresponding author. ah.haghighi@gmail.com, Tel: +98(919)3996782.

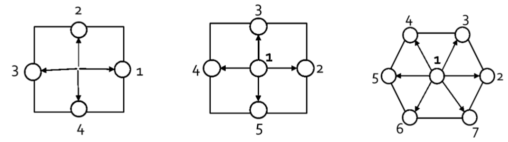
†Department of Mathematics, Technical and Vocational University, Tehran, Iran.

‡Department of Mathematic, Urmia University of Technology, Urmia, Iran.

§Department of Mathematic, Urmia University of Technology, Urmia, Iran.

physics, chemical physics and nonlinear physics, which use high-dimensional forms of this equation [14]-[17]. The Burgers equation was first introduced by Bateman [18]. He solved this equation in its one-dimensional form using a simple solution procedure. Later, Burgers used the Batmans method as a simplified model to interpret the theory of turbulence [19]. Using the one-dimensional form of this equation, Cole [20] and Hopf [21] found that it can represent the typical features of the shock wave theory. Miller [22] presented an analytical solution for Burgers equation using the series solution. Due to its tremendous applications mentioned above, many numerical approaches have been developed to solve this equation such as the finite element method [23], moving finite element scheme [24], mixed finite element technique [25], finite difference method [26]-[28], Chebyshev spectral collocation methods [29], collocation procedures using cubic B-spline [30] and Adomian decomposition method [31]. The local Crank-Nicolson (LCN) and the modified local Crank-Nicolson (MLCN) methods were introduced by Abduwali [32, 33] for the solution of heat conduction equation. Huang and Abduwali [34] solved the one and two-dimensional Burgers equations using the MLCN method. The MLCN method leads to several block matrices through the transformation of the partial differential equation (PDE) into ordinary differential equations (ODE) and thus simplifies the calculations. The salient feature of the MLCN method is the fact that it solves the equations directly without using any transformations like the Hopf-Cole transformation. The conventional Crank-Nicolson method is an implicit method, which is based on the central difference in space and the trapezoidal rule in time giving second-order convergence in time [35]. The MLCN transforms the PDE into several ODEs and uses the Trotter product formula of the exponential for the approximation of the coefficient matrix of these ODEs. It also separates the coefficient matrix into some small-block matrices. In addition, the MLCN method employs the conventional Crank-Nicolson method to advance the solution in time. Therefore, unlike the conventional Crank-Nicolson method the MLCN is explicit and unconditionally stable. The Lattice Boltzmann method (LBM) can be regarded either as an extension of the lattice gas cellular automata (LGCA) or as a special discrete form of the Boltzmann equation for the kinetic theory [36].

Unlike the conventional numerical methods, which are based on the discretization of PDEs describing macroscopic conservation laws, the LBM is based on the solution of the discrete-velocity Boltzmann equation. It describes the microscopic picture of particles movements in an extremely simplified way, while on the macroscopic level it gives a correct average description [37, 38]. The LBM has been developed to simulate both linear and nonlinear PDEs such as wave equation [39], Burgers equation [40]-[43], Korteweg de Vries (KdV) equation [44] and Lorenz equation [45]. In [41], the LBM was developed for the solution of the one-dimensional Burgers equation with a 2-bit model where using the Taylor expansion and multi-scale analysis, the modified Burgers equation was recovered from the Lattice Boltzmann equations (LBEs), and the local equilibrium distribution functions were obtained. Gao [46] used a 3-bit model for the solution of the Burgers equation, where the errors were found to be smaller than those in Ref [39]. Duan and Liu [47] solved the two-dimensional Burgers equation with the LBM (4-bit) model. Zhang and Yan [48] used the LBM (5-bit) and the LBM (7-bit) for the solution of one and two-dimensional Burgers equations. Although there have been several methods for solving the Burgers equations (as mentioned above), to the authors best of knowledge, a comprehensive comparison between the accuracy of these methods especially at different Reynolds numbers has not been carried out. As such, in this paper, the two-dimensional Burgers equations are solved using different LBM models and the MLCN method for different Reynolds numbers, the results are compared with the exact solution counterparts and the accuracy of these methods are evaluated.



(a) a (b) b (c) c

Figure 1: a.b Diagrammatic sketch of squares lattice, c. regular hexagon

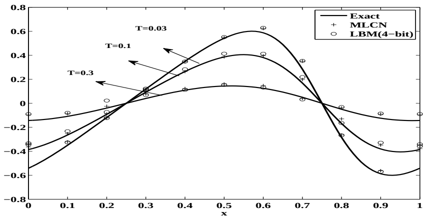
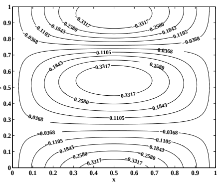
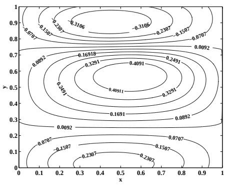


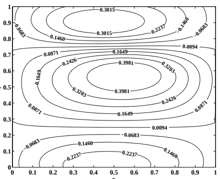
Figure 2: Profile of u for exampleI for T=0.03,T=0.1,T=0.3 and Δt=0.0013 and Re=10.



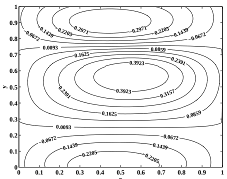
(a) Exact solution



(b) LBM(4-bit) model

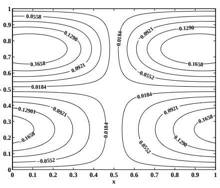


(c) LBM(5-bit) model

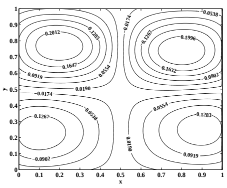


(d) LBM(7-bit) model

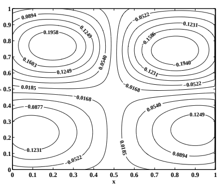
Figure 3: Contours of u for Example I of Case I, (a) Contours of the exact solution result, (b) Contours of the LBM(4-bit) result, (c) Contours of the LBM(5-bit) result and (d) Contours of the LBM(7-bit) result.



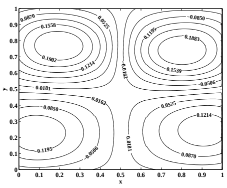
(a) Exact solution



(b) LBM(4-bit) model

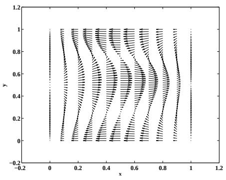


(c) LBM(5-bit) model

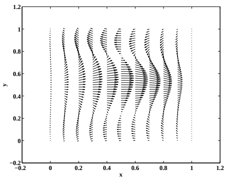


(d) LBM(7-bit) model

Figure 4: Contours of v for Example I of Case I, (a) Contours of the exact solution result, (b) Contours of the LBM(4-bit) result, (c) Contours of the LBM(5-bit) result and (d) Contours of the LBM(7-bit) result.

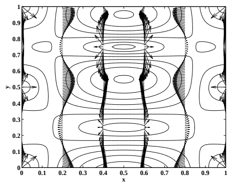


(a) Exact solution

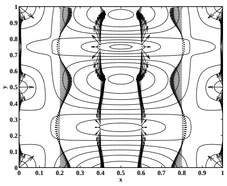


(b) LBM(7-bit) model

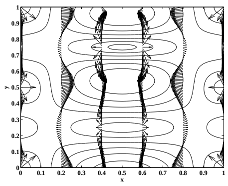
Figure 5: Velocity vectors of (u, v) for Example I of Case I, (a) for the exact solution result, (b) for the LBM(7-bit) result.



(a) Exact solution

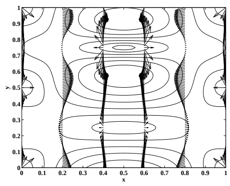


(b) MLCN method

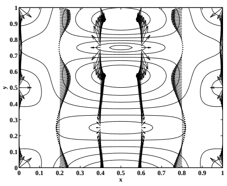


(c) LBM(7-bit) model

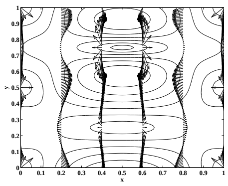
Figure 6: Contours of $V = \sqrt{u^2 + v^2}$ for Example I of Case I, (a) Contours of the exact solution result, (b) Contours of the MLCN method, (c) Contours of the LBM(7-bit) result.



(a) Exact solution

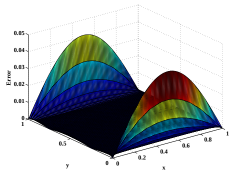


(b) MLCN method

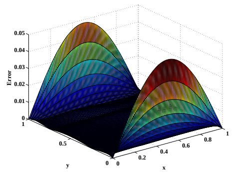


(c) LBM(7-bit) model

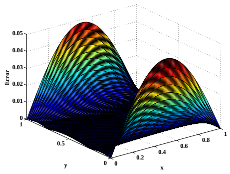
Figure 7: Contours of $V = \sqrt{u^2 + v^2}$ for Example I of Case II, (a) Contours of the exact solution result, (b) Contours of the MLCN method, (c) Contours of the LBM(7-bit) result.



(a) T=0.03

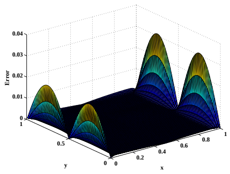


(b) T=0.1

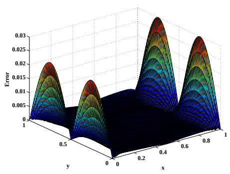


(c) T=0.3

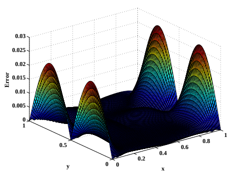
Figure 8: Average error of $|u_{exact} - u_{LBM(7-bit)}|$ for Example I at different Time steps and Re=10 and $\Delta t = 0.0013$, (a) T=0.03, (b) T=0.1 and (c) T=0.3.



(a) T=0.03

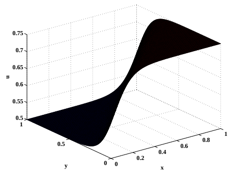


(b) T=0.1

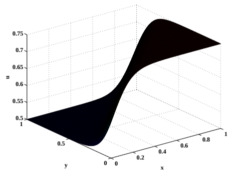


(c) T=0.3

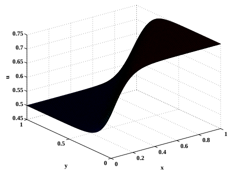
Figure 9: Average error of $|v_{exact} - v_{LBM(7-bit)}|$ for Example I at different Time steps and Re=10 and $\Delta t = 0.0013$, (a) T=0.03, (b) T=0.1 and (c) T=0.3.



(a) Exact solution

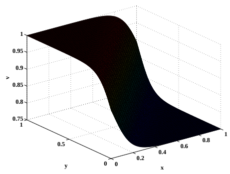


(b) LBM(7-bit) model

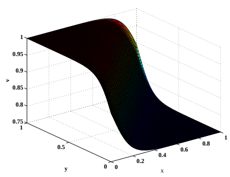


(c) MLCN method

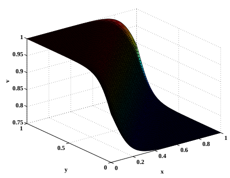
Figure 10: Comparison of u for Example II of CaseII, (a) profile of the exact solution result, (b) profile of the LBM(7-bit) result and (c) profile of the MLCN method result.



(a) Exact solution



(b) LBM(7-bit) model



(c) MLCN method

Figure 11: Comparison of v for Example II of CaseII, (a) profile of the exact solution result, (b) profile of the LBM(7-bit) result and (c) profile of the MLCN method result.

$$A_{11}=\begin{pmatrix} -8\nu & 2\nu-hv_{11}^n & 0 & \cdots & 0 & 2\nu-hv_{11}^n & 0 & \cdots & 0\\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0\\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots\\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

$$A_{1i}=\begin{pmatrix} 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0\\ \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots\\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0\\ 0 & \cdots & 0 & 2\nu+hv_{1i}^n & -8\nu & 2\nu-hv_{1i}^n & 0 & \cdots & 0 & 2\nu-hv_{1i}^n & 0 & \cdots & 0\\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0\\ \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots\\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

$$A_{1,M-1}=\begin{pmatrix} 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0\\ \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots\\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0\\ 0 & \cdots & 0 & 2\nu+hv_{1,M-1}^n & -8\nu & 0 & 0 & \cdots & 0 & 2\nu-hv_{1,M-1}^n & 0 & \cdots & 0\\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0\\ \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots\\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

$$A_{i1}=\begin{pmatrix} 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0\\ \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots\\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0\\ 2\nu+hv_{i1}^n & 0 & \cdots & 0 & -8\nu & 2\nu+hv_{i1}^n & 0 & \cdots & 0 & 2\nu-hv_{i1}^n & 0 & \cdots & 0\\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0\\ \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots\\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

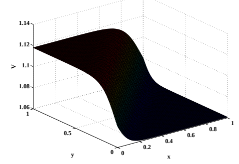
$$A_{ii}=\begin{pmatrix} 0 & \cdots & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \cdots & 0\\ \vdots & & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & & \vdots\\ 0 & \cdots & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \cdots & 0\\ 0 & \cdots & 2\nu+hv_{ii}^n & \cdots & 0 & 2\nu+hv_{ii}^n & -8\nu & 2\nu-hv_{ii}^n & 0 & \cdots & 2\nu-hv_{ii}^n & \cdots & 0\\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0\\ \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots\\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

$$A_{i,M-1}=\begin{pmatrix} 0 & \cdots & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 2\nu+hv^n_{i,M-1} & \cdots & 0 & 2\nu+hv^n_{i,M-1} & -8\nu & 0 & \cdots & 2\nu-hv^n_{i,M-1} & \cdots & 0 & \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

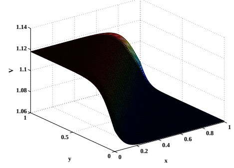
$$A_{M-1,1}=\begin{pmatrix} 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 2\nu+hv^n_{M-1,1} & 0 & \cdots & 0 & -8\nu & 2\nu-hv^n_{M-1,1} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

$$A_{M-1,i}=\begin{pmatrix} 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 2\nu+hv^n_{M-1,i} & 0 & \cdots & 0 & 2\nu-hv^n_{M-1,i} & -8\nu & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

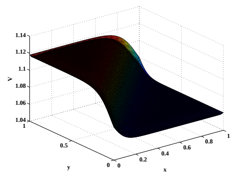
$$A_{M-1,M-1}=\begin{pmatrix} 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \cdots & 0 & 2\nu+hv^n_{M-1,M-1} & 0 & \cdots & 0 & 2\nu+hv^n_{M-1,M-1} & -8\nu \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix}$$



(a) Exact solution



(b) LBM(7-bit) model



(c) MLCN method

Figure 12: Comparison of $V=\sqrt{u^2+v^2}$ for Example II of CaseII, (a) profile of the exact solution result, (b) profile of the LBM(7-bit) result and (c) profile of the MLCN method result.

2 Two-dimensional Burgers equations

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+u\frac{\partial u}{\partial y}=\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right), \quad (2.1)$$

$$\frac{\partial v}{\partial t}+v\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=\frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right), \quad (2.2)$$

$$(x,y)\in\Omega, \quad \tau\in[0,T]$$

with the initial conditions

$$\begin{aligned} u(x,y,t)&=u_0(x,y), \quad (x,y)\in(0,T] \\ v(x,y,t)&=v_0(x,y), \quad (x,y)\in(0,T] \end{aligned} \quad (2.3)$$

and the boundary conditions

$$\begin{aligned} u&=0, \quad (x,y)\in\partial\Omega \quad t\in(0,T] \\ v&=0, \quad (x,y)\in\partial\Omega \quad t\in(0,T] \end{aligned} \quad (2.4)$$

where Re is Reynolds number and u_0 is a given function. The Burgers Eqs. 2.1 and 2.2 with the Re term behaves as a parabolic and without it behaves merely as a hyperbolic PDE and the problem becomes very difficult to solve as a steep shock-like wave front developes [34], [34], [36].

3 LBM for solving the Burgers equation

According to the theory of the LBM, it consists of two steps: (1) streaming, where each particle moves to the nearest node in the direction of its velocity and (2) collision, which occurs when particles arriving at a node interact and possibly change their velocity directions according to scattering rules. These two steps can be combined into the following LBE:

$$f_\alpha(x+e_\alpha\Delta t,t+\Delta t)-f_\alpha(x+t)= \frac{1}{\tau}\left(f_\alpha-f_\alpha^{eq}\right) \quad (3.5)$$

where f_α is the distribution function of particles, f_α^{eq} is the local equilibrium function of particles, Δx and Δt are space and time increments, respectively; $c=\Delta x/\Delta t$ is the speed of light in the system and e_α is the velocity vector of a particle in the direction and τ is the dimensionless relaxation time, which controls the rate of approach to equilibrium state.The macroscopic velocity u is defined in terms of the distribution functions as

$$u=\sum_\alpha f_\alpha=\sum_\alpha f_\alpha^{eq}=\sum_\alpha f_\alpha^{(0)} \quad (3.6)$$

3.1 LBM(4-bit) model

In this model, the square lattice is used , (see Fig. 1). To derive the macroscopic properties from the lattice BGK model, the Taylor expansion and multi scale analysis are used. The distribution functions are expanded up to linear terms in a small expansion parameter ϵ,

$$f_\alpha=f_\alpha^{(0)}+\epsilon f_\alpha^{(0)}+o\left(\epsilon^2\right).$$

From the kinetic Eq. 3.5, the distribution function
$f_\alpha(x+\Delta t e_\alpha,t+\Delta t)$ is expanded using the Taylor
expansion and the approximation of $f_\alpha^{(1)}$ is calculated,

$$\begin{aligned} &f_\alpha(x+e_\alpha\Delta t,t+\Delta t)= \\ &f_\alpha(x,t)+\Delta e_{\alpha_i}\partial_{x_i}f_\alpha+\Delta t\partial_t f_\alpha+o\left(\epsilon^2\right)= \\ &\left(1-\frac{1}{\tau}\right)f_\alpha(x,t)+\frac{1}{\tau}f_\alpha^{eq}(x,t)\,\epsilon f_\alpha^{(1)}= \\ &-\tau\Delta t\left(\partial_t+e_\alpha\partial_x\right)f_\alpha \end{aligned} \quad (3.7)$$

Table 1: Profile of u for example I. for $T = 0.1, \Delta t = 0.0013$ and $Re = 10$

(x,y)	Exact	MLCN	$LBM_{(4)}$	$LBM_{(5)}$	$LBM_{(7)}$	$\mid E \mid_{(MLCN)}$	$\mid E \mid_{(4)}$	$\mid E \mid_{(5)}$	$\mid E \mid_{(7)}$
$(0.1, 0.1)$	-0.091	-0.087	-0.095	-0.094	-0.093	0.004	0.003	0.002	0.002
$(0.5, 0.1)$	0.119	0.110	0.126	0.125	-0.122	0.009	0.007	0.006	0.031
$(0.9, 0.1)$	-0.102	-0.094	-0.111	-0.110	-0.107	0.008	0.008	0.007	0.005
$(0.3, 0.3)$	0.078	0.076	0.078	0.078	0.078	0.001	0.004	0.003	0.000
$(0.7, 0.3)$	0.126	0.118	0.134	0.132	0.123	0.008	0.082	0.006	0.002
$(0.1, 0.5)$	-0.264	0.241	-0.241	-0.240	-0.249	0.023	0.022	0.023	0.015
$(0.5, 0.5)$	0.385	0.376	-0.394	-0.392	0.389	0.009	0.008	0.006	0.003
$(0.9, 0.5)$	-0.380	-0.371	-0.372	-0.374	-0.376	0.009	0.008	0.006	0.004
$(0.3, 0.7)$	0.078	0.081	0.086	0.083	0.081	0.003	0.008	0.005	0.003
$(0.7, 0.7)$	0.126	0.132	0.137	0.136	0.131	0.006	0.011	0.009	0.005
$(0.1, 0.9)$	-0.091	0.084	-0.081	-0.079	-0.083	0.007	0.007	0.012	0.008
$(0.5, 0.9)$	0.119	0.120	0.136	0.134	0.132	0.001	0.017	0.015	0.013
$(0.9, 0.9)$	-0.102	-0.102	-0.109	-0.108	-0.105	0.002	0.007	0.0059	0.002
$E_{ave} = \frac{1}{N}\sum\lvert E\rvert$						0.007	0.009	0.007	0.004

Table 2: Profile of $\sqrt{u^2+v^2}$ for exampleI. for $T = 0.1, \Delta t = 0.0013$ and $Re = 10$

(x)	Exact	$LBM_{(4)}$	$LBM_{(5)}$	$LBM_{(7)}$	$\mid E \mid_{(4)}$	$\mid E \mid_{(5)}$	$\mid E \mid_{(7)}$
$0.1, 0.1$	0.1371	0.1198	0.1200	0.1232	0.0173	0.0171	0.0139
$0.5, 0.1$	0.1192	0.1102	0.1262	0.1279	0.0090	0.0070	0.0087
$0.9, 0.1$	0.1532	0.1596	0.1589	0.1561	0.0064	0.0057	0.0292
$0.3, 0.3$	0.1170	0.1116	0.1132	0.1144	0.0054	0.0038	0.0026
$0.7, 0.3$	0.1895	0.1183	0.2123	0.1328	0.0712	0.0672	0.0567
$0.1, 0.5$	0.2644	0.1599	0.1670	0.2408	0.1045	0.0974	0.0236
$0.5, 0.5$	0.3859	0.3989	0.3976	0.3871	0.0130	0.0117	0.0012
$0.9, 0.5$	0.3809	0.2712	0.2788	0.3241	0.1097	0.1021	0.0568
$0.3, 0.7$	0.1170	0.1213	0.1208	0.1178	0.0043	0.0038	0.0008
$0.7, 0.7$	0.1895	0.2513	0.2469	0.2162	0.0618	0.0574	0.0267
$0.1, 0.9$	0.1371	0.0742	0.0814	0.0987	0.0629	0.0110	0.0004
$0.5, 0.9$	0.1192	0.1342	0.1368	0.1209	0.0176	0.0150	0.0017
$0.9, 0.9$	0.1532	0.1120	0.1286	0.1381	0.0412	0.0246	0.0150
$E_{ave} = \frac{1}{N}\sum\lvert E\rvert$					0.0403	0.0326	0.0161

By introducing time scale $t_2 = \epsilon^2 t$, and space scale $x_1 = \epsilon x$, then time derivation and the space derivation can be expanded formally:

$$\partial_t = \epsilon^2 \partial_{t_2}, \quad \partial_x = \epsilon \partial_{x_1} \tag{3.8}$$

Substituting relations 3.8 into the equation 3.7 and neglecting the terms of order $o\left(\epsilon^3\right)$, a multi-scale equation is obtained,

$$\begin{aligned} &\Delta t \epsilon^2 \partial_{t_2} \textstyle\sum_\alpha f_\alpha^{(0)} + \Delta t \epsilon \partial_{x_1} \sum_\alpha e_\alpha f_\alpha^{(0)} \\ &+\Delta^2 \epsilon^2 \left(\frac{1}{2} - \tau\right) \partial x_1 \partial x_1 e_\alpha e_\alpha \textstyle\sum_\alpha f_\alpha^{(0)} \\ &= 0. \end{aligned} \tag{3.9}$$

Corresponding to the macroscopic Eqs. 2.1 and 2.2, we let

$$\sum_\alpha e_{\alpha_i} f_\alpha^{(0)} = \frac{u^2}{2}, \tag{3.10}$$

$$\sum_\alpha e_{\alpha_i} e_{\alpha_j} f_\alpha^{(0)} = \eta u \delta_{ij}. \tag{3.11}$$

Using Eqs 3.6 and 3.7, the equilibrium distribution function $f_\alpha^{(0)}$ and η function in Eq. 3.10 are

Table 3: Profile of u for exampleI. for $T = 0.1, \Delta t = 0.0013$ and $Re = 100$

(x, y)	Exact	MLCN	$LBM_{(4)}$	$LBM_{(5)}$	$LBM_{(7)}$	$\mid E \mid_{(MLCN)}$	$\mid E \mid_{(4)}$	$\mid E \mid_{(5)}$	$\mid E \mid_{(7)}$
$(0.1, 0.1)$	-0.0137	-0.0127	-0.0125	-0.0133	-0.0133	0.0010	0.0012	0.0004	0.0004
$(0.5, 0.1)$	0.0184	0.0155	0.0177	0.0181	-0.0128	0.0029	0.0007	0.0003	0.0002
$(0.9, 0.1)$	-0.0163	-0.0126	-0.0157	-0.0165	-0.0163	0.0037	0.0006	0.0002	0.0001
$(0.3, 0.3)$	0.0109	0.0108	0.0107	0.0108	0.0108	0.0001	0.0002	0.0001	0.0002
$(0.7, 0.3)$	0.0235	0.0137	0.0223	0.0233	0.0233	0.0098	0.0002	0.0001	0.0001
$(0.1, 0.5)$	-0.0367	-0.0372	-0.0346	-0.0364	-0.0366	0.0005	0.0021	0.0003	0.0001
$(0.5, 0.5)$	0.0597	0.0401	0.0584	0.0594	0.0598	0.0001	0.0013	0.0005	0.0001
$(0.9, 0.5)$	-0.0670	-0.0495	-0.0608	-0.0647	-0.0650	0.0175	0.0062	0.0023	0.0020
$(0.3, 0.7)$	0.0109	0.0108	0.0108	0.0109	0.0110	0.0001	0.0001	0.0001	0.0001
$(0.7, 0.7)$	0.0235	0.0113	0.0229	0.0238	0.0237	0.0122	0.0006	0.0003	0.0002
$(0.1, 0.9)$	-0.0137	-0.0124	-0.0122	-0.0129	-0.0129	0.0013	0.0015	0.0008	0.0007
$(0.5, 0.9)$	0.0184	0.0159	0.0185	0.0188	0.0188	0.0025	0.0001	0.0004	0.0003
$(0.9, 0.9)$	-0.0163	-0.0123	-0.0150	-0.0159	-0.0159	0.0040	0.0013	0.0004	0.0005
$E_{ave} = \frac{1}{N}\sum \lvert E \rvert$						0.0058	0.0012	0.0005	0.0003

Table 4: Profile of $\sqrt{u^2 + v^2}$ for exampleI. for $T = 0.1, \Delta t = 0.0013$ and $Re = 100$

(x, y)	Exact	$LBM_{(4)}$	$LBM_{(5)}$	$LBM_{(7)}$	$\mid E \mid_{(4)}$	$\mid E \mid_{(5)}$	$\mid E \mid_{(7)}$
$(0.1, 0.1)$	0.0207	0.0203	0.0204	0.0206	0.0004	0.0003	0.0001
$(0.5, 0.1)$	0.0185	0.0182	0.0183	0.0184	0.0003	0.0002	0.0001
$(0.9, 0.1)$	0.0246	0.0250	0.0251	0.0253	0.0007	0.0005	0.0003
$(0.3, 0.3)$	0.0164	0.0160	0.0162	0.0163	0.0004	0.0002	0.0001
$(0.7, 0.3)$	0.0354	0.0358	0.0356	0.0353	0.0004	0.0002	0.0001
$(0.1, 0.5)$	0.0378	0.0370	0.0371	0.0375	0.0008	0.0006	0.0003
$(0.5, 0.5)$	0.0598	0.0602	0.0596	0.0599	0.0004	0.0002	0.0001
$(0.9, 0.5)$	0.0672	0.0658	0.0664	0.0671	0.0014	0.0008	0.0001
$(0.3, 0.7)$	0.0164	0.0167	0.0165	0.0163	0.0003	0.0002	0.0001
$(0.7, 0.7)$	0.0354	0.0364	0.0362	0.0360	0.0010	0.0008	0.0006
$(0.1, 0.9)$	0.0207	0.0197	0.0198	0.0200	0.0010	0.0009	0.0007
$(0.5, 0.9)$	0.0185	0.0189	0.0188	0.0186	0.0004	0.0003	0.0001
$(0.9, 0.9)$	0.0246	0.0240	0.0242	0.0244	0.0006	0.0004	0.0002
$E_{ave} = \frac{1}{N}\sum \lvert E \rvert$					0.0006	0.0004	0.0002

obtained as,

$$f_\alpha^{eq} = \begin{cases} \dfrac{u^2}{2} + \dfrac{u^2}{4c}, & \alpha = 1, 2 \\ \\ \dfrac{u^2}{2} - \dfrac{u^2}{4c}, & \alpha = 3, 4 \end{cases} \quad , \quad \eta = c^2 \tag{3.12}$$

So the viscosity $\nu = 1/Re$ is defined by

$$\nu = \eta \left(\tau - \frac{1}{2}\right) \Delta = \frac{1}{2} \left(\tau - \frac{1}{2}\right) \Delta t c^2$$

Stability of this method has been proved in Ref. [28].

3.2 LBM(5-bit) model

In this model, the square lattice is used (see Fig. 1).
The equilibrium distribution function of this model is given as,

$$f_0^{(0)} = \left(1 - \frac{2\eta}{c^2} u\right) - \frac{2u^3}{3c^2}$$

$$f_1^{(0)} = \frac{\eta u}{2c^2} + \frac{u^2}{4c} + \frac{u^3}{6c^2}$$

Table 5: Profile of u for exampleI. for $\Delta t = 0.0013$ and $Re = 100$

(x, y)	T=0.03			T=0.1			T=0.3		
	Exact	$LBM_{(7)}$	$\mid E \mid_{(7)}$	Exact	$LBM_{(7)}$	$\mid E \mid_{(7)}$	Exact	$LBM_{(7)}$	$\mid E \mid_{(7)}$
$(0.1, 0.1)$	-0.014	-0.014	0.001	-0.013	-0.013	0.002	-0.010	-0.010	0.004
$(0.5, 0.1)$	0.019	0.019	0.001	0.018	0.018	0.003	0.016	0.016	0.004
$(0.9, 0.1)$	-0.017	-0.017	0.002	-0.016	-0.016	0.004	-0.013	-0.014	0.001
$(0.3, 0.3)$	0.011	0.011	0.001	0.010	0.010	0.001	0.010	0.010	0.002
$(0.7, 0.3)$	0.025	0.024	0.001	0.023	0.023	0.001	0.020	0.020	0.002
$(0.1, 0.5)$	-0.038	-0.038	0.001	-0.037	-0.037	0.004	-0.027	-0.029	0.001
$(0.5, 0.5)$	0.062	0.061	0.001	0.059	0.059	0.004	0.054	0.054	0.002
$(0.9, 0.5)$	-0.070	-0.070	0.001	-0.066	-0.066	0.004	-0.048	-0.049	0.001
$(0.3, 0.7)$	0.011	0.011	0.001	0.011	0.010	0.001	0.010	0.010	0.001
$(0.7, 0.7)$	0.025	0.024	0.002	0.024	0.023	0.006	0.021	0.020	0.001
$(0.1, 0.9)$	-0.014	-0.014	0.001	-0.013	-0.013	0.005	-0.009	-0.010	0.001
$(0.5, 0.9)$	0.019	-0.019	0.002	0.018	0.018	0.004	0.017	0.016	0.009
$(0.9, 0.9)$	-0.017	-0.017	0.001	-0.016	-0.016	0.002	-0.012	-0.013	0.001
$E_{ave} = \frac{1}{N}\sum\lvert E\rvert$			0.001			0.003			0.00

Table 6: Profile of u for exampleII. for $T = 0.1, \Delta t = 0.0013$ and $Re = 10$

(x, y)	Exact	$LBM_{(7-bit)}$	MLCN	$\mid E \mid_{(7-bit)}$	$\mid E \mid_{(MLCN)}$
$(0.1, 0.1)$	0.6231	0.6072	0.6012	0.0159	0.0219
$(0.5, 0.1)$	0.5926	0.5821	0.5787	0.0105	0.0139
$(0.9, 0.1)$	0.5657	0.5623	0.5612	0.0034	0.0045
$(0.3, 0.3)$	0.6231	0.6072	0.6012	0.0159	0.0219
$(0.7, 0.3)$	0.5926	0.5821	0.5787	0.0105	0.0139
$(0.1, 0.5)$	0.6538	0.6378	0.6291	0.0160	0.0247
$(0.5, 0.5)$	0.6231	0.6072	0.6012	0.0159	0.0219
$(0.9, 0.5)$	0.5926	0.5821	0.5787	0.0105	0.0139
$(0.3, 0.7)$	0.6538	0.6278	0.6291	0.0160	0.0247
$(0.7, 0.7)$	0.6231	0.6072	0.6012	0.0159	0.0219
$(0.1, 0.9)$	0.6812	0.6712	0.6686	0.0100	0.0126
$(0.5, 0.9)$	0.6538	0.6278	0.6291	0.0160	0.0247
$(0.9, 0.9)$	0.6231	0.6072	0.6012	0.0159	0.0219
$E_{ave} = \frac{1}{N}\sum\lvert E\rvert$				0.0132	0.0186

$$f_2^{(0)} = \frac{\eta u}{2c^2} + \frac{u^2}{4c} + \frac{u^3}{6c^2}$$

$$f_3^{(0)} = \frac{\eta u}{2c^2} - \frac{u^2}{4c} + \frac{u^3}{6c^2}$$

$$f_4^{(0)} = \frac{\eta u}{2c^2} - \frac{u^2}{4c} + \frac{u^3}{6c^2}$$

Stability of this method has been proved in Ref. [48].

3.3 LBM(7-bit) model

In this model, the regular hexagon is used (see Fig. 1).

The equilibrium distribution function of this model is given as,

$$f_0^{(0)} = \left(1 - \frac{2\eta}{c^2}u\right) - \frac{2u^3}{3c^2},$$

$$f_1^{(0)} = \frac{\eta u}{3c^2} + \frac{2}{3}\left(\frac{u^2}{4c}\right) + \frac{u^3}{9c^2},$$

$$f_2^{(0)} = \frac{\eta u}{3c^2} + \left(\frac{3+\sqrt{3}}{3} - \frac{2}{3}\right)\frac{u^2}{4c} + \left(\sqrt{3}+1\right)\frac{u^3}{6c^2},$$

$$f_3^{(0)} = \frac{\eta u}{3c^2} - \left(\frac{3+\sqrt{3}}{3} - \frac{2}{3}\right)\frac{u^2}{4c} + \left(-\sqrt{3}+1\right)\frac{u^3}{6c^2},$$

Table 7: Profile of v for exampleII. for $T = 0.1, \Delta t = 0.0013$ and $Re = 10$.

(x, y)	Exact	$LBM_{(7-bit)}$	MLCN	$\mid E \mid_{(7-bit)}$	$\mid E \mid_{(MLCN)}$
$(0.1, 0.1)$	0.8769	0.8681	0.8631	0.0088	0.0138
$(0.5, 0.1)$	0.9074	0.8994	0.8926	0.0080	0.0148
$(0.9, 0.1)$	0.9343	0.9286	0.9257	0.0057	0.0086
$(0.3, 0.3)$	0.8769	0.8681	0.8631	0.0088	0.0138
$(0.7, 0.3)$	0.9074	0.8994	0.8926	0.0080	0.0148
$(0.1, 0.5)$	0.8462	0.8413	0.8378	0.0049	0.0084
$(0.5, 0.5)$	0.8769	0.8761	0.8631	0.0088	0.0138
$(0.9, 0.5)$	0.9074	0.8994	0.8926	0.0080	0.0148
$(0.3, 0.7)$	0.8462	0.8413	0.8378	0.0049	0.0084
$(0.7, 0.7)$	0.8769	0.8681	0.8631	0.0088	0.0138
$(0.1, 0.9)$	0.8188	0.8115	0.8094	0.0073	0.0094
$(0.5, 0.9)$	0.8462	0.8413	0.8378	0.0049	0.0084
$(0.9, 0.9)$	0.8769	0.8681	0.8631	0.0088	0.0138
$E_{ave} = \frac{1}{N}\sum\lvert E\rvert$				0.0074	0.0120

Table 8: Profile of v for exampleII. for $T = 0.1, \Delta t = 0.0013$ and $Re = 10$.

(x, y)	Exact	$LBM_{(7-bit)}$	$LBM_{(MLCN)}$	$\mid E \mid_{(7-bit)}$	$\mid E \mid_{(MLCN)}$
$(0.1, 0.1)$	1.0757	1.0712	1.0681	0.0045	0.0076
$(0.5, 0.1)$	1.0838	1.0791	1.0713	0.0047	0.0125
$(0.9, 0.1)$	1.0922	1.0872	1.0805	0.0050	0.0117
$(0.3, 0.3)$	1.0757	1.0712	1.0681	0.0045	0.0076
$(0.7, 0.3)$	1.0838	1.0791	1.0713	0.0047	0.0125
$(0.1, 0.5)$	1.0694	1.0622	1.0794	0.0072	0.0100
$(0.5, 0.5)$	1.0757	1.0712	1.0681	0.0045	0.0076
$(0.9, 0.5)$	1.0838	1.0791	1.0713	0.0047	0.0125
$(0.3, 0.7)$	1.0694	1.0622	1.0794	0.0072	0.0100
$(0.7, 0.7)$	1.0757	1.0712	1.0681	0.0045	0.0076
$(0.1, 0.9)$	1.0838	1.0791	1.0723	0.0047	0.0125
$(0.5, 0.9)$	0.1192	1.1082	1.1015	0.0110	0.0177
$(0.9, 0.9)$	0.1532	0.1481	0.1415	0.0051	0.0117
$E_{ave} = \frac{1}{N}\sum\lvert E\rvert$				0.0056	0.0109

$$f_4^{(0)} = \frac{\eta u}{3c^2} - \frac{2}{3}\left(\frac{u^2}{4c}\right) + \frac{u^3}{9c^2},$$

$$f_5^{(0)} = \frac{\eta u}{3c^2} - \left(\frac{3+\sqrt{3}}{3} - \frac{2}{3}\right)\frac{u^2}{4c} + \left(\sqrt{3}+1\right)\frac{u^3}{6c^2},$$

$$f_6^{(0)} = \frac{\eta u}{3c^2} + \left(\frac{3-\sqrt{3}}{3} - \frac{2}{3}\right)\frac{u^2}{4c} + \left(-\sqrt{3}+1\right)\frac{u^3}{6c^2}.$$

Stability of this method has been proved in Ref. [48].

4 MLCN method for solving the Burgers' equation

using the central difference quotient method to discretize the Eqs. 2.1 and 2.2, the following semi-discrete equation is obtained,

$$\frac{dV(t)}{dt} = \frac{1}{2h^2} AV(t) \tag{4.13}$$

where $V(t)$ is a vector that expresses an approximate solution of u into Eqs. 2.1 and 2.2. Let h be the mesh width in space, Δt be the mesh width in time and A is a $(M-1)^2 \times (M-1)^2$ block tridiagonal matrix. So by integrating Eqs. 2.1 and

Table 9: Profile of u for exampleII. for $T = 0.1, \Delta t = 0.0013$ and $Re = 100$.

(x, y)	Exact	$LBM_{(7-bit)}$	MLCN	$\mid E \mid_{(7-bit)}$	$\mid E \mid_{(MLCN)}$
$(0.1, 0.1)$	0.6059	0.6186	0.6234	0.0127	0.0175
$(0.5, 0.1)$	0.5007	0.5022	0.5012	0.0005	0.0010
$(0.9, 0.1)$	0.5000	0.5001	0.4985	0.0001	0.0015
$(0.3, 0.3)$	0.6059	0.6251	0.6251	0.0192	0.0192
$(0.7, 0.3)$	0.5012	0.5022	0.5023	0.0010	0.0011
$(0.1, 0.5)$	0.7477	0.7478	0.7435	0.0001	0.0042
$(0.5, 0.5)$	0.6059	0.6251	0.6251	0.0192	0.0192
$(0.9, 0.5)$	0.5012	0.5022	0.5022	0.0010	0.0010
$(0.3, 0.7)$	0.7477	0.7478	0.7479	0.0001	0.0002
$(0.7, 0.7)$	0.6059	0.6251	0.6251	0.0008	0.0008
$(0.1, 0.9)$	0.7500	0.7500	0.7475	0.0001	0.0025
$(0.5, 0.9)$	0.7477	0.7478	0.7479	0.0001	0.0002
$(0.9, 0.9)$	0.6059	0.6251	0.6251	0.0192	0.0192
$E_{ave} = \frac{1}{N}\sum \lvert E \rvert$				0.0057	0.0067

Table 10: Profile of v for exampleII. for $T = 0.1, \Delta t = 0.0013$ and $Re = 100$.

(x, y)	Exact	$LBM_{(7-bit)}$	MLCN	$\mid E \mid_{(7-bit)}$	$\mid E \mid_{(MLCN)}$
$(0.1, 0.1)$	0.8941	0.8891	0.8815	0.0050	0.0126
$(0.5, 0.1)$	0.9988	0.9927	0.9892	0.0061	0.0096
$(0.9, 0.1)$	1.0000	0.9948	0.9913	0.0052	0.0081
$(0.3, 0.3)$	0.8941	0.8912	0.8842	0.0029	0.0099
$(0.7, 0.3)$	0.9988	0.9977	0.9977	0.0011	0.0011
$(0.1, 0.5)$	0.7253	0.7497	0.7480	0.0026	0.0043
$(0.5, 0.5)$	0.8941	0.8912	0.8842	0.0029	0.0099
$(0.9, 0.5)$	0.9988	0.9977	0.9977	0.0011	0.0011
$(0.3, 0.7)$	0.7523	0.7522	0.7521	0.0001	0.0002
$(0.7, 0.7)$	0.8941	0.8912	0.8842	0.0029	0.0099
$(0.1, 0.9)$	0.7500	0.7475	0.7458	0.0025	0.0042
$(0.5, 0.9)$	0.7523	0.7522	0.7521	0.0001	0.0002
$(0.9, 0.9)$	0.8941	0.8912	0.8842	0.0029	0.0099
$E_{ave} = \frac{1}{N}\sum \lvert E \rvert$				0.0027	0.0062

2.2, and with the vector of $V(t_n)$, we have

$$V(t_{n+1}) = exp\left(\frac{\Delta t}{2h^2}A\right)V(t_n). \tag{4.14}$$

Consider the Crank-Nicolson scheme for Eqs. 2.1 and 2.2

$$V(t_{n+1}) = \left((1-\lambda A)^{-1}((1+\lambda A)\right)V(t_n) \tag{4.15}$$

where, $\lambda = \tau/4h^2$ is the ratio mesh.
By comparing Eqs. 4.13, and 4.14, we obtain the approximation as follows:

$$exp\left(\frac{\Delta t}{2h^2}A\right) \simeq (1-\lambda A)^{-1}(1+\lambda A) \tag{4.16}$$

To solve Eq. 4.14, we must obtain an approximation for $exp\left(\frac{\Delta t}{2h^2}\right)A$.
Using the Trotter Product formula, we deduce an iterative formula as follows,

$$exp\left(\frac{\Delta t}{2h^2}A_{ij}\right) \simeq \prod_{i,j}^{M-1}(1-\lambda A_{ij})^{-1}(1+\lambda A_{i,j}) \tag{4.17}$$

where A_{ij} are split of matrix A , and we have,

Table 11: Profile of $\sqrt{u^2+v^2}$ for example II. for $T=0.1, \Delta t=0.0013$ and $Re=100$.

(x,y)	Exact	$LBM_{(7-bit)}$	MLCN	$\mid E\mid_{(7-bit)}$	$\mid E\mid_{(MLCN)}$
$(0.1,0.1)$	1.0801	1.0712	1.0651	0.0089	0.0150
$(0.5,0.1)$	1.1175	1.1089	1.1035	0.0086	0.0140
$(0.9,0.1)$	1.1180	1.1082	0.1012	0.0098	0.0168
$(0.3,0.3)$	1.0801	1.0758	1.0754	0.0043	0.0047
$(0.7,0.3)$	1.1175	1.1172	1.1170	0.0003	0.0005
$(0.1,0.5)$	1.0607	1.0526	1.0512	0.0081	0.0095
$(0.5,0.5)$	1.0801	1.0758	1.0754	0.0043	0.0047
$(0.9,0.5)$	1.1175	1.1089	1.1035	0.0086	0.0140
$(0.3,0.7)$	1.0607	1.0608	1.0609	0.0001	0.0002
$(0.7,0.7)$	1.0801	1.0758	1.0754	0.0043	0.0047
$(0.1,0.9)$	1.0607	1.0526	1.0512	0.0081	0.0095
$(0.5,0.9)$	1.0607	1.0608	1.0605	0.0001	0.0002
$(0.9,0.9)$	1.0801	1.0758	1.0754	0.0043	0.0047
$E_{ave}=\frac{1}{N}\sum\lvert E\rvert$				0.0054	0.0076

Then applying 4.14 and 4.16, we see that

$$V\left(t_{n+1}\right)=\prod_{i,j=1}^{M-1}\left(\left(1-\lambda A_{ij}\right)^{-1}\left(\left(1+\lambda A_{ij}\right)\right)V\left(t_n\right)$$

In order to improve the numerical accuracy, we can write,

$$V_1\left(t_{n+1}\right)=\prod_{i,j=1}^{M-1}\left(\left(1-\lambda A_{ij}\right)^{-1}\left(\left(1+\lambda A_{ij}\right)\right)V_1\left(t_n\right)$$

$$V_2\left(t_{n+1}\right)=\prod_{i,j=1}^{M-1}\left(\left(1-\lambda B_{ij}\right)^{-1}\left(\left(1+\lambda B_{ij}\right)\right)V_2\left(t_n\right)$$

where, $B_{ij}=A_{M-i,M-j}$.
Stability of this method has been proved in Ref. [34].

5 Results and Discussion

In this paper, the two-dimensional Burgers Eqs. 2.1 and 2.2 with different initial conditions are studied to verify the accuracy of numerical methods of MLCN and LBM. Through solving two examples, the results of LBM and MLCN are compared with the exact solution to determine which one is more accurate. For this purpose, Example I is solved with MLCN and LBM(4-bit) and the results are compared with the exact solution. Since the accuracy of the MLCN is dependent on the mesh size Δx and the accuracy of the LBM method is dependent on the mesh size Δx as well as the number of particle velocities, so the effects of these two important parameters are considered. The most accurate method is used to solve Example II. The results of MLCN method and LBM are considered for different Re numbers and different time steps.

5.1 Example I

In this Example, we consider the system of two-dimensional Burgers equation given in Eqs. 2.1 and 2.2, over a square domain $D:[0,1]\times[0,1]$, with the initial conditions,

$$u\left(x,y,0\right)=\frac{-4\nu\pi cos\left(2\pi x\right)sin\left(\pi y\right)}{2+sin\left(2\pi x\right)sin\left(\pi y\right)},$$
$$v\left(x,y,0\right)=\frac{-2\nu\pi sin\left(2\pi x\right)cos\left(\pi y\right)}{2+sin\left(2\pi x\right)sin\left(\pi y\right)},$$

$(x,y)\in D$, and boundary conditions

$$\begin{aligned}
&u\left(0,y,t\right)=-2\nu\pi\exp^{-5\pi^2\nu t}sin\left(\pi y\right),\\
&u\left(1,y,t\right)=-2\nu\pi\exp^{-5\Pi^2\nu t}sin\left(\pi y\right),\\
&u\left(x,0,t\right)=0,\quad t\geq 0,\\
&u\left(x,1,t\right)=0,\quad t\geq 0,\\
&v\left(0,y,t\right)=0,\quad t\geq 0,\\
&v\left(1,y,t\right)=0,\quad t\geq 0,\\
&v\left(x,0,t\right)=-\nu\pi\exp^{-5\pi^2\nu t}sin\left(2\pi x\right),t\geq 0,\\
&v\left(x,1,t\right)=\nu\pi\exp^{-5\pi^2\nu t}sin\left(2\pi x\right),t\geq 0.
\end{aligned}$$

5.1.1 Exact solution

The exact solution of Eqs. 2.1 and 2.2 can be found by using the Hopf-Cole transformation [49],

$$u(x,y,t) = -2\nu \frac{\pi \exp^{-5\pi^2 \nu t} cos(2\pi x)\, sin(\pi y)}{2 + \exp^{-5\pi^2 \nu t} sin(2\pi x)\, sin(\pi y)},$$

$$v(x,y,t) = -2\nu \frac{\pi \exp^{-5\pi^2 \nu t} sin(2\pi x)\, cos(\pi y)}{2 + \exp^{-5\pi^2 \nu t} sin(2\pi x)\, sin(\pi y)}.$$

5.1.2 Effect of Δx on the accuracy of the result

As previously mentioned, the spatial accuracy of MLCN and LBM is dependent on the mesh size Δx [50, 51]. In order to show this, several simulations are performed using different grid sizes and the results of LBM and MLCN are compared with the exact solution. Through a mesh-independent study the optimum grid size Δx was obtained to be $\Delta x = 1/50$ and $\Delta x = 1/100$ for the MLCN method and the LBM method, respectively. In Fig. 2, we compare the results associated to the MLCN method and the LBM (4-bit) method with the exact solution. Figure 2 shows the values of x-component velocity u at y=0.5 obtained with optimum Δx. Other parameters are $T = 0.03, T = 0.1, T = 0.3$ and $Re = 10, \Delta t = 0.0013$. It can be seen from Fig. 2 that the results of the MLCN are closer to those of the exact solution as compared with those of the LBM method. This may be attributed to the small number of particle velocity directions (4-bit) used in the LBM.

5.1.3 Effect of number of particle velocity directions

Case I: $Re = 10, \Delta t = 0.0013$ and T=0.1
In this section, the contours of velocity components u and v are ploted. Figures 3 and 4 show the velocity contours of u and v, respectively at time T=0.1 and Re=10 for difeerent numbers of particle velocity directions. Figure 3a shows the exact solution and Figs. 3b to 3d show the the results of LBM(4-bit), LBM(5-bit) and LBM(7-bit), respectively. The same trend is the case for Fig. 4. The negative and positive values of contour lables represent the direction of the velocity components in the computaional domian.

The velocity vectors are plotted in Fig. 5 for better understanding of the flow behaviour in the domein. Figures 5a and 5b show the velocity vectors associated respectively with the exact solution and the LBM(7-bit) at Re=10 indicating good agrrement between them. As can be seen from Table 1, it is clear that results of LBM (7-bit) for Re=10 have the best agreement with result of Exact solution because of higher number of particle velocity directions.

In order to compare the results of LBM (7-bit) with MLCN, the results associated with some grid points are given in Table 1 and their average error are reported. It can be seen that the results are in good agreement with the exact solution. However, the LBM (7-bit) method gives the most accurate results (the lowest average error) amongst the other LBM methods having different particle velocity directions and is more accurate than the MLCN method. In Table 2, the values of velocity magnitude $|\overrightarrow{V}| = V = \sqrt{u^2+v^2}$ for Re=10 are given for Exact solution and LBM with different numbers of particle velocity and MLCN, respectively. It is observed from this table that the LBM(7-bit) gives the best accuracy as compared with LBM(4-bit), LBM(5-bit) and MLCN. In Fig 6a-c the velocity vectors and contours are depicted for Exact solution, LBM(7-bit) and MLCN, respectively. In the following section, the Re number is increased to prove the priority of LBM(7-bit) with respect to the other methods employed in the present work.

5.1.4 The accuracy of the the LBM and the MLCN methods at various Reynolds numbers

For the optimum value of Δx obtained for the MLCN method and the LBM, the results are compared with the exact solution for two different Re numbers to determine which one is more accurate. In order to do this, we consider in this section another test case as Case II where all the conditions are held the same as for the Case I (see section 5.1.3) except Re number, which is taken to be 100.

Case II: Re=100, $\Delta t = 0.0013$,T=0.1
As mentioned, for this Case II all the conditions are held the same as for the Case I but the Re number is taken to be 100. For this Case II, the values of x-component velocity u are shown in Table 3. It is observed from Tables 1 and 3 that

for both Re=10 and Re=100, the solutions of the LBM are more accurate than those of the MLCN method. As may be evident, when the Re number is increased the value of Δx should be decreased to achieve the desired accuracy. This may be due to the fact that by increasing the Re number the boundary layer thickness becomes smaller, meaning that the velocity gradients are more appreciable in the boundary layer. It is worth mentioning that through a mesh-independent study the optimum grid size Δx for Case II was obtained to be 1/100 and 1/200 for the MLCN method and the LBM method, respectively. The results for this case are given in table 3 from comparison of Case I and Case II (see Table 1 and Table 3), we found that the accuracy of the LBM is improved by increasing the Re number. Also the average error is less than that of the Case I. however, no significant changes occur in the MLCN results with respect to the Case I.

In Table 2 and 4 velocity values V are given for Re=10 and Re=100, respectively. It is observed from these Tables that at higher Re number, LBM(7-bit) has a smaller average error compared with LBM(4-bit), LBM(5-bit) and MLCN method. Also, it can be seen that, the LBM method results at higher Re numbers gives more accurate.

Figures 6 and 7 depict the velocity vectors and contours associated respectively Re=10 and Re=100. in these figures the results associated with the exact solution, LBM(7-bit) and MLCN methods are compared. It can be seen that both the LBM and MLCN give rise to reasonably accurate results, although it is hard to identify the more accrate method from these figures. in order to find more accurate method, we show in Tables 3 and 4 the velocity values V for Re=10 and Re=100, Respectively. It is observed from this tables that, with increasing Re number (Re=100), LBM(7-bit) has a less average error in compared with LBM(4-bit), LBM(5-bit) and MLCN method. Also can be seen that with increasing Re number accuracy of LBM (7-bit) is less than accuracy of LBM (7-bit) in Re=10.

We obtain the average error associated to the LBM (7-bit) and exact solution at different Time steps. Figures 8 and 9 show error=$|u_{exact} - u_{LBM(7-bit)}|$ associated respectively Re=10 and Re=100. other parametres are T=0.03, T=0.1 and T=0.3 and $\Delta t = 0.0013$. In order to compare the results of LBM (7-bit) with exact solution, the results associated with some grid points at different time steps (T=0.03, T=0.1 and T=0.3) and Re=100, are given in Table 5 and the average error are reported.

5.2 Example II

In this Example, we consider the system of two-dimensional Burgers equation given in Eqs. 2.1 and 2.2, over a square domain $D : [0, 1] \times [0, 1]$, with the exact solutions [52]:

$$u\left(x,y,t\right) = \frac{3}{4} - \frac{1}{4\left(1 + exp\left(Re\dfrac{(4y-4x-t)}{32}\right)\right)},$$

$$v\left(x,y,t\right) = \frac{3}{4} + \frac{1}{4\left(1 + exp\left(Re\dfrac{(4y-4x-t)}{32}\right)\right)}.$$

The macroscopic initial condition and boundary conditions are determined by the exact solution.

The main difference between this Example II and Example I is due to the difference in the initial conditions. It is mentioned in Example I that, the LBM (7-bit) is the most accurate velocity model. So the results associated with this Example II are obtained using LBM (7-bit). The other paprameters are considered the same as those in Example I. Similarly, the results are obtained for two different Re numbers of 10 and 100 and the results of MLCN method and LBM are compared with the exact solution.

5.2.1 The accuracy of the the LBM and the MLCN methods at various Re numbers

In Tables 6-8, we compare the results of the MLCN method, the LBM (7-bit) and the exact solution associated with Example II. For specified values of x-component velocity u, y-component velocity v and x-component velocity V for Re=10 are given for exact solution, the MLCN method and the LBM (7-bit) method. It can be seen from these Tables that the results of the LBM are

closer to those of the exact solution, compared with those of the MLCN method. This example is solved for case II to check the accuracy of the numerical methods at different Re numbers. As previously mentioned, since by increasing the Re number value of Δx should be decreased to achieve the best accuracy. As been obtained, through a mesh-independent study the optimum grid size Δx was obtained to be $\Delta x = 1/100$ and $\Delta x = 1/200$ for the MLCN method and the LBM method, respectively.
The results for Case II are shown in Tables 9-11 for specified values of velocity u, v and V, respectively. We have seen from these Tables that the results of the LBM are Better accuracy from the MLCN method. We can found from the results of case I and case II, that average error for case II is less than case I. Figs 10-12 show the values velocity u, v and V, respectively, that obtained with optimum Δx and Re=100 for case II.

So in this example, the accuracy of the LBM is better than the MLCN method at high Re number.

6 Conclusion

The Burgers equation is a combination of a convection term and a diffusion term which is simplified form of the Navier-Stokes equation. The modified local Crank-Nicolson and Lattice Boltzmann methods for the two-dimensional Burgers equations have been presented and the results of these methods are compared with the result from the exact solution. It is shown that the method has an explicit difference. Scheme with unconditionally stability. In support of the given method, two test examples have been solved with different initial conditions and the accuracy of the MLCN method and the LBM is considered in different Re numbers. We have found that the results of the LBM (7-bit) are better than the LBM (4-bit) and LBM (5-bit) also, the results are much better for larger values of Re number. Results show that at high Re numbers the accuracy of the LBM is more than the MLCN method.

References

[1] S. Dhawan, Contemporary review of techniques for the solution of nonlinear Burgers' equation, *Journal of Computational Science* 5 (2012) 405-419.

[2] S. N. Gurbatov, A. N. Malakhov, A. I. Saichev, Nonlinear random waves and turbulence in nondispersive media: waves, rays, particles, *Manchester University Press* 5 (1991) 405-419.

[3] M. Kardar, G. Parisi, Y. C. Zhang, Dynamic scaling of growing interfaces, *Physical Review Letters* 9 (1986) 889-892.

[4] M. Vergassola, Burgers' equation, Devil's staircases and the mass distribution for large-scale structures, *Astronomy, Astrophysics* 289 (1994) 325-356.

[5] N. Collier, Time adaptivity in the diffusive wave approximation to the shallow water equations, *Journal of Computational Science* 3 (2013) 152-156.

[6] S. A. Molchanov, D. Surgailis, W. A. Woyczynski, Hyperbolic asymptotics in Burgers' turbulence and extremal processes, *Communications in Mathematical Physics* 1 (1995) 209-226.

[7] G. Adomian, The diffusion-Brusselator equation, *Computers . Mathematics with Applications* 5 (1995) 1-3.

[8] B. E. Clothier, J. H. Knight, I. White, Burgers' equation: Application to Field Constant-Flux Infiltration, *Soil Sci.* 132 (1981) 99-261.

[9] E. Demetriou, M. A. Christou, C. Sophocleous, On the classification of similarity solutions of a two-dimensional diffusion-advection equation, *Applied Mathematics* 187 (2007) 1333-1350.

[10] L. Rajaee, H. Eshraghi, R. O. Popovych, Multi-dimensional guasi-simple wave in weakly dissipative flows, *Physica D: Nonlinear Phenomena* 237 (2008) 405-419.

[11] L. A. Richards, Capillary conduction of liquids through porous mediums, *Physics* 5 (1931).

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

[13] S. Esipov, Coupled Burgers equations: a model of polydispersive sedimentation, *Phys* 10 (1995).

[14] M. P. Edwards, P. Broadbridge, Exceptional symmerty reductions of Burgers' equation in two and three spatial dimensions, *Zeitshrift fur angewandte Mathematik and physick ZAMP* 46 (1995) 595-622.

[15] H. Hasegawa, Plasma Instabilities and nonlinear Effects, *Sppringer-Verlag* (1975).

[16] G. Gray, Chemical Oscillations and Instabilities Clarendon, *Oxford* (1990)

[17] M. A. Christou, C. Sophocleous, Numerical similarity reductions of the (1+3)-dimensional Burgers equation, *Applied Mathematics and Computation* 18 (2011) 7455-7461.

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

[19] L. Lighthill, Viscosity effects in sound waves of finite amplitude, in: G.K. Batchlor, *Surveys in Mechanics* 250 (1956).

[20] J. D. Cole, On a Quasi-Linear Parabolic Equation Occuring in Aerodynamics, *Quart. Appl. Math.* 4 (1951) 225-236.

[21] E. Hopf, The partial differential equation $u_t + uu_x = \mu xx$, *Communications on Pure and Applied Mathematics* (1950) 201-230.

[22] E. L. Miller, Predictor-corrector studies of Burgers' model of turbulent flow, (1966).

[23] J. Caldwell, P. Wanless, A. E. Cook, A finite element approach to Burgers' equation, *Appl. Math. Mod.* 3 (1981) 189-193.

[24] A. N. Hrymak, G. J. McRae, A. W. Westerberg, An implementation of a moving finite element method, *J. Comp. Phy.* 1 (1986) 168-190.

[25] H. Chen, Z. Jiang, A characteristics-mixed finite element method for Burgers' equation, *Korean J. Comput. Appl. Math* 2 (2004) 29-51.

[26] D. J. Evans, A. R. Abdullah, The Group Explicit method for the solution of Burger's equation, *Computing* 3 (1984) 239-253.

[27] S. Kutluay, A. R. Bahadir, A. zde, Numerical solution of one-dimensional Burgers equation: explicit and exact-explicit finite difference methods, *J. Compu. Appl. Math.* 2 (1984) 251-256.

[28] A. H. Tabatabaei, E. Shakour, M. Dehghan, Some implicit methods for the numerical solution of Burgers equation, *App. Math. Compu.* 2 (2007b) 560-570.

[29] N. Bressan, A. Quarteroni, An implicit/explicit spectral method for Burgers' equation, *CALCOLO* (1986) 565-284.

[30] A. Ali, G. A. Gardner, L. Gardner, A collocation solution for Burgers' equation using cubic B-spline finite elements, *Comput. Methods Appl. Mech. Eng.* 3 (1992b) 325-337.

[31] A. R. Haghighi, M. Shojaeifard, Numerical solution of the one dimensional nonlinear Burgers equation using the Adomian decomposition method and the comparison between the modified Local Crank-Nicolson method and the VIM exact solution, *Int. j. of Industrial mathematics* 2 (2015) 149-159.

[32] A. Abuduwali, M. Sakakihara, H. Niki, A local Crank-Nicolson method for solving the heat equation, *Comput. Methods Appl. Mech. Eng* (1994) 1-13.

[33] A. Abuduwali, A corrector local C-N method for the two-dimensional heat equation, *Math. Comp. Mod.* 19 (1997).

[34] P. Huang, A. Abduwali, The Modified Local Crank Nicolson method for one- and two-dimensional Burgers equations *Comp. Math. with Appl.* 8 (2010b) 2452-2463.

[35] A. R. Haghighi, A. Shahbazi, A comparison between alternating segment Crank-Nicolson and explicit-implicit schemes for the dispersive equation, *Int. J. Comp. Sci. Math.* 4 (2014) 405-417.

[36] A. R. Haghighi, S. Pakrou, Comparison of the LBM with the modified local Crank-Nicolson method solution of transient one-dimensional nonlinear Burgers' equation, *Int. J. Comp. Sci. Math* 5 (2016) 459-466.

[37] R. Benzi, M. Vergassola, The lattice Boltzmann equation: theory and Applications, *Rep, Phys* 222 (1992) 145-97.

[38] S. Chen, G. Doolen, Lattice boltzmann method for fluid flows , *Annual Review of Fluid Mechanics* 1 (1998) 329-364.

[39] Y. Guangwu, A Lattice Boltzmann Equation for Wave, *J. Compu. Physics* 161 (2000) 61-69.

[40] B. M. Boghosian, J. Yepez, Entropic lattice Boltzmann model for Burgers equation, *Phil. Trans. Roy. Soc, Lon. A* 362 (2004) 1691-1702.

[41] Y. Duan, R. Liu, Y. Jiang, Lattice Boltzmann model for the modified Burgers equation, *Appl. Math. Comp.* 2 (2008) 489-497.

[42] L. Huilin, A new lattice Boltzmann model for solving the coupled viscous Burgers' equation, *Physica A* (2014) 445-457.

[43] L. Qianhuan, C. Zhenhua, S. Baochang, A novel lattice Boltzmann model for the coupled viscous Burgers' equations, *Appl. Math. Comp.* 250 (2015) 948-957.

[44] J. Zhang, G. Yan, A lattice Boltzmann model for the Korteweg de Vries equation with two conservation laws, *Computer Physics Communications* 7 (2009) 1054-1062.

[45] G. W. Yan, Lattice Bhatnager-Groos-Krook model for the Lorenz attractor, *Physica D* 154 (2007) 43-50.

[46] Y. Gao, B. C. Shi, Numerical solution of Burgers equation by lattice Boltzmann method, *Applied Mathematics and Computation* 219 (2013) 7685-7692.

[47] Y. Duan, R. Liu, Y. Jiang, Lattice Boltzmann model for two-dimensional unsteady Burgers equation, *J. of Comput. and Appl. Math.* 206 (2007) 432-439.

[48] Z. Jianging Zhang, Y. Guangwu, Lattice Boltzmann method for one and two-dimensional Burgers equations, *Physica A* 387 (2011) 4771-4786.

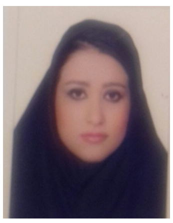
[49] W. Y. Liao, A fourth order finite-difference method for solving the system of two-dimensional Burgers' equations, *Int. J. Number. Methods Fluids* 64 (2008) 565-590.

[50] A. R. Haghighi, N. Aliashrafi, N. Asghary, An implicit finite difference scheme for analyzing the effect of body acceleration on pulsatile blood flow through a stenosed artery, *J. of Linear and Topological Algebram* 2 (2017) 147-161.

[51] A. R, Dadvand, A. Haghighi, H. Hoseini, Numerical simulation of natural convection heat transfer in a horizontal annulus with discrete heating using mesh-free lattice-Boltzmann method, *Advances In Mathematical Modeling* 5 (2015) 89-112.

[52] R. C. Mittal, R. Jiwari, Differential quadrature method for two-dimentional Burgers equation, *Int. J. Comput. Methods Eng. Sci. Mech.* 10 (2009) 450-459.

Ahmad Reza Haghighi is an Associate Professor in the department of mathematics at Technical and Vocational University(TVU), Tehran, Iran. He completed his Ph. D degree in applied mathematics from Pune University, India. His research interest includes Bio-Mathematics, Computational Fluid Dynamics, Partial Differential Equations and Control.



Shirin Pakrou was born in Salmas, Iran. She received her B.Sc. degree in applied mathematics from University of Mohaghegh Ardabili. Also, she holds master degree in applied mathematics from Urmia University of Technology, Urmia, Iran. Her research areas include Bio-Mathematics, Numerical analysis, Partial Differential Equations.

Abdol Rahman Dadvand is an Associate Professor in the Department of Mechanical Engineering Urmia University of Technology, Urmia,Iran. He completed his Ph. D degree in Mechanical Engineering (Energy Conversion), (2009) University of Tabriz, Tabriz, Iran. His research interest includes Computational fluid dynamics (CFD).