Libyan Journal of Science & Technology 13:1 (2021) 48–56

Faculty of Science - University of Benghazi

Libyan Journal of Science & Technology

journal home page: www.sc.uob.edu.ly/pages/page/77

The totally volume local discontinuous Galerkin method for viscous Burgers' equations.

Elhadi I. Elhadi[a], Mouad A. Fakroon[b,*], Abdelhadi A. Buzghaiba[b]

[a] *Department of Mechanical Engineering, Faculty of Engineering, University of Benghazi, Benghazi, Libya*

[b] *Department of Aeronautical Engineering, Faculty of Aeronautical Sciences, University of Bright Star, Berega, Libya*

Highlights

- **The present method has a better agreement with the analytical solution in comparison with the other existing numerical solutions.**
- **The proposed scheme treats effectively with strong discontinuities without producing nonphysical oscillations.**
- **Errors are decreasing as the number of elements is increased.**

ARTICLE INFO

Article history:

Received 09 November 2020
Revised 10 June 2021
Accepted 13 June 2021

Keywords:

Burgers' equation; Totally volume local discontinuous Galerkin method; strong stability preserving Runge Kutta explicit method.

**Address of correspondence:*

E-mail address: fakroonmuaz@gmail.com

M. A. Fakroon

ABSTRACT

In this paper, the totally volume local discontinuous Galerkin TV-LDG method is proposed to solve the viscous Burgers' equations with appropriate initial and boundary conditions. Time derivative is discretized by the third-order strong stability preserving Runge Kutta explicit SSP-RK (3, 3) method. Space derivatives discretization is performed by the totally volume local discontinuous Galerkin method. Finally, the validity of proposed scheme is demonstrated by numerical experiments and shows that the present scheme offers better accuracy in comparison with other existing numerical methods.

1. Introduction

Problems of practical interest in which convection and diffusion play an important role arise in applications as diverse as weather-forecasting, problems of environmental pollution, oceanography, gas dynamics, aeroacoustics, oil recovery simulation, modeling of shallow water, transport of contaminant in fluids, semiconductor device simulation, among many others which could be described by partial differential equations. one of the well-known partial differential equations employing to govern convection-diffusion processes are viscous Burgers' equations. This is why devising robust, accurate, and efficient methods for numerically solving these partial differential equations is of considerable importance and, as expected, has attracted the interest of many researchers.

Viscous Burgers' equation is a non-linear conservation equation. The numerical solution of the Navier–Stokes equations is a challenging problem for computational fluid dynamics that requires careful mathematical and numerical formulation. As a simplified model of the Navier-Stokes equations, the viscous Burgers' equation represents many of the properties of Navier-Stokes equations, such as non-linear convection and viscous diffusion, leading to shock wave formation and boundary layers. Viscous Burgers' equation is used in computational fluid dynamics as a simplified model for turbulence, boundary layer behavior, shock wave formation, and mass transport.

Consider two-dimensional viscous Burgers' equations:

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=\alpha\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right) \qquad (1)$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=\alpha\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right) \qquad (2)$$

on the domain interval $\Omega=\{(x,y)|a\leq(x,y)\geq b\},\ t>0.$

where $u(x,y,t)$and $v(x,y,t)$ are transported variables and α is the diffusion coefficient.

The Burgers' equation was first introduced by Bateman (1915). It was later referred to as the Burgers' equation after Burger (1948) introduced this equation as a mathematical model for fluid flow. Various powerful mathematical methods such as Cole-Hopf transformation (Taku, 2009), the-function method (Soliman, 2006), variational iteration method (Abdou and Soliman, 2005), and other methods have been used to solve one-dimensional viscous Burgers' equation analytically. Solving the two dimensional viscous Burgers' equations numerically is a natural first step towards developing methods for the computation of complex flows. Many researchers have developed various numerical schemes for solving the Burgers' equations to validate their algorithm. These numerical methods include implicit methods (Bahadir, 1999), the boundary element method (Bahadir, 2005). In addition, a fully finite-difference scheme has been introduced by Bahadir (2003) to obtain the numerical solution of (2D) viscous Burgers' equations. Mittal and Jiwari (2009) applied a differential quadrature method to solve the two dimensional viscous Burgers' equations. The Local

 ISSN: Online 2663-1407, Print 2663-1393; National Library of Libya, Legal number: 390/2018

discontinuous Galerkin LDG method has been introduced (Cockburn and Shu, 1998) to deal with nonlinear convection-diffusion equation containing viscous terms.

In this work, we illustrate the essential ideas of the LDG method and how we can transform high order partial differential equations into a system of first-order partial differential equations by introducing a new auxiliary variable q to approximate the derivative of the solution V.

The main objective of this study is to develop the LDG by using the divergence theorem to unify the integrals of the governing equation (boundary integral and volume integral). The unified integrals and the local solvability of all the auxiliary variables are why the present method is called the totally volume integral local discontinues Galerkin TV-LDG method. Since the first-order system of equations will discretize by using the TV-LDG space discretization method. Then the obtained system of ordinary differential equations will be integrated in time by using the strong stability preserving Runge-Kutta SSP-RK third-order time discretization method (Shu and Osher, 1988).

2. The TV-LDG Method

Consider the two-dimensional time-dependent linear convection-diffusion equation:

$$\frac{\partial V}{\partial t}+\frac{\partial f(V)}{\partial x}+\frac{\partial f(V)}{\partial y}=\alpha\left[\frac{\partial^2 V}{\partial x^2}+\frac{\partial^2 V}{\partial y^2}\right] \tag{3}$$

where $f(V)=\frac{V^2}{2}$ is the convection flux of the vector V of the scalar components u and v.

By introducing a new auxiliary variable $q=\alpha(\nabla.V)$ we can rewrite Eq. (3) as a system of first-order equations:

$$\frac{\partial V}{\partial t}+\nabla.f(V)-\nabla.q=0 \tag{4}$$

$$q-\alpha(\nabla.V)=0 \tag{5}$$

assuming we are solving these systems of Eq. (4) and Eq. (5) on interval $\Omega\in[a,b]$. We divide the domain $[a,b]$ into equally space N elements.

First of all, the whole domain Ω is divided into small computational cells $\Omega=\cup_{j=1}^{N}\Omega_j$, where Ω_j is the subdomain called cell or element, the length of the cell for the one-dimensional domain is $h=\Delta x=[x_{j+\frac{1}{2}}-x_{j-\frac{1}{2}}]$, in case of two dimensional the mesh size, is $h=\Delta x=\Delta y$, then spatial discretization of the first-order system is performed by TV-LDG method, the weak forms of the system of Eq. (3) and Eq. (4) are obtained by the scalar multiplication of the partial differential equations with test functions w and p then the integration by parts is applied over the subdomain Ω_j.

Discrete analogues of Eq. (3) and Eq. (4) are obtained by considering V_h, q_h, w and p within each element defined as:

$$V(x,y,t)_h=\sum_{i=1}^{N_j}V_i(t)\phi_i^k(x,y),$$

$$q(x,y)_h=\sum_{i=1}^{N_j}q_i\,\phi_i^k(x,y),$$

$$w(x,y)_h=\sum_{i=1}^{N_j}w_i\,\phi_i^k(x,y),$$

$$p(x,y)_h=\sum_{i=1}^{N_j}p_i\,\phi_i^k(x,y),$$

where the expansion coefficients $V_i(t), q_i, w_i$ and p_i denote the degrees of freedom of the numerical solution and of the test function in element Ω_j, and the N_j (shape) functions ϕ_i^k are the basis of the Lagrange polynomials P_k.

$$\int_{\Omega_j} w\frac{\partial V_h}{\partial t}d\Omega_j-\int_{\Omega_j}\nabla w(f(V_h)-q_h)d\Omega_j+\oint_{\partial\Omega_j} w(\hat{f}-\hat{q})\cdot n\,d\partial\Omega_j=0 \quad (5)$$

$$\int_{\Omega_j} pq_h\,d\Omega_j+\alpha\int_{\Omega_j}\nabla p\,V_h d\Omega_j-\alpha\oint_{\partial\Omega_j}(p\hat{V})\cdot n\,d\partial\Omega_j=0 \tag{6}$$

where $\partial\Omega_j$ the boundary of the element and n is the unit outward normal vector to the boundary and, all the "hat" terms $\hat{f}$, $\hat{q}$ and $\hat{V}$ are the numerical fluxes that designed to approximate the convective and diffusion fluxes at the boundaries of the element $\partial\Omega_j$. The total volume integral of the numerical fluxes (Elhadi *et al.*, 2020) is used to unify the integrals (boundary integral and volume integral) in Eq. (5) and Eq. (6). Hence the obtained equations can be written as:

$$\int_{\Omega_j}[w\frac{\partial V_h}{\partial t}+\nabla w\,(f(V_h)-q_h)+\nabla(w\,(\hat{f}-\hat{q}))]d\Omega_j=0 \tag{7}$$

$$\int_{\Omega_j}\left[p\,q_h+(\nabla p)\,V_h-\nabla\left(p\,\hat{V}\right)\right]d\Omega_j=0 \tag{8}$$

Now the numerical flux $\hat{V}$ is the approximation of V_h on the element boundaries and depend on the solution of both sides of the element interface $\hat{V}\left(V_{j+\frac{1}{2}}^{+},V_{j+\frac{1}{2}}^{-}\right)$, where the $V_{j+\frac{1}{2}}^{+}$ and $V_{j+\frac{1}{2}}^{-}$ are the values of V_h at $x_{j+\frac{1}{2}}$ from the right element Ω_{j+1}, and the left element Ω_j, respectively.

In this research work, there are two types of numerical fluxes to be defined. Firstly, the diffusion numerical fluxes $\hat{V}$ and $\hat{q}$, secondly the convection numerical flux $\hat{f}$. Lax-Friederichs fluxes (Toro, 1999) is used for the convection numerical flux:

$$\hat{f}=\frac{1}{2}\left(f(V^-)+f(V^+)-\delta(V^+-V^-)\right)$$

where δ is the maximum absolute value of the eigenvalues of the Jacobian matrix. The diffusion numerical fluxes $\hat{V}$ and $\hat{q}$ can be chosen as central fluxes (Bassi and Rebay, 1997):

$$\hat{V}_{j+\frac{1}{2}}=\frac{1}{2}(V_h^{+}+V_h^{-})_{j+\frac{1}{2}} \tag{9}$$

$$\hat{q}_{j+\frac{1}{2}}=\frac{1}{2}(q_h^{+}+q_h^{-})_{j+\frac{1}{2}} \tag{10}$$

by applying the numerical integration and assembling all the elemental contributions, the system of ordinary differential equations that govern the evolution in time of the discrete solution can be written as:

$$M\frac{du_h}{dt}=R(u_h) \tag{11}$$

where M is the mass matrix obtained after applying the numerical integration over the cell and u_h is the global vector of the degrees of freedom $R(u_h)$ is the residual of the process resulting from Eq. (7) and Eq. (8).

1-

3. Time Integration

The main idea of the totally volume Integral local discontinuous Galerkin TV-LDG method is that the auxiliary variable q can be solved explicitly and locally (in element Ω_j) in terms of u_h by inverting the element mass matrix inside the cell Ω_j. Thus, the elimination of the auxiliary variable q has been done in the Eq. (8), then obtain the combined ordinary differential equation system for freedoms U_h as follows:

$$\frac{d}{dt}U_h=M^{-1}R(u_h)=L(U_h,t) \tag{12}$$

where this ordinary differential equation appears from the discretization of the spatial derivative in the partial differential equation. This semi-discretize scheme is discretized in time by using the third order strong stability preserving Runge-Kutta SSP-RK method (Shu and Osher, 1988), where U^n is the solution at the time t^n and the solution at the next time step is U^{n+1} which is obtained after the s stages, where the time marching algorithm performs by using the three-stage third-order Runge–Kutta method as follows:

$$U^{(1)} = U^n + \Delta t \cdot L(U^n, t^n) \tag{13}$$

$$U^{(2)} = \frac{3}{4}U^n + \frac{1}{4}U^{(1)} + \frac{1}{4}\Delta t \cdot L\left(U^{(1)}, t^n + \Delta t\right) \tag{14}$$

$$U^{(3)} = \frac{1}{3}U^n + \frac{2}{3}U^{(2)} + \frac{2}{3}\Delta t \cdot L\left(U^{(2)}, t^n + \frac{1}{2}\Delta t\right) \tag{15}$$

4. Numerical Results

In order to illustrate the performance of the proposed scheme for solving the viscous Burgers' equations and justifying the accuracy and efficiency of the TV-LDG method, we considered two test examples. To show the efficiency of the present method for our problem as compared with the exact solution, we report a maximum error, which is defined as:

$$L_\infty = \max\left\|V_{i,h} - V_{i,exact}\right\|$$

where $V_{i,exact}$ is the exact solution and $V_{i,h}$ is the numerical solution obtained by the present method at every node in the domain.

4.1. Test Example 1

The first problem is the one dimensional viscous Burgers' equation to test the present scheme when it deals with nonlinear time-dependent convection-diffusion problems:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \alpha\frac{\partial^2 u}{\partial x^2} \tag{16}$$

with an initial condition as:

$$u(x, 0) = u_0(x) = sin(\pi x)$$

the boundary conditions:

$$u(0, t) = u(2, t) = 0, \ t > 0.$$

the exact solution of this problem is:

$$u(x, t) = \frac{2\pi\alpha \sum_{n=1}^{\infty} a_n \, exp\{-n^2\pi^2\alpha t\} n sin(n\pi x)}{a_0 + \sum_{n=1}^{\infty} a_n \, exp\{-n^2\pi^2\alpha t\} cos(n\pi x)}$$

where the Fourier coefficient are

$$a_0 = \int_0^2 exp\{-(2\pi\alpha)^{-1}[1 - \cos(\pi x)]\} dx$$

$$a_n = 2\int_0^2 exp\{-(2\pi\alpha)^{-1}[1 - \cos(n\pi x)]\} \cos(n\pi x) dx$$

The problem domain [0, 2] is divided into 40 equally spaced, in the following calculation the diffusion coefficient α is taken as (1, 0.1 and 0.01) for the linear element $k = 1$ and the quadratic element $k = 2$, respectively. Comparisons are made with the exact solution and numerical solutions of several existing numerical schemes, which are fully implicit finite difference method IFDM (Bahadir, 1999), boundary element method BEM (Bahadir, 2005), and the numerical results, are presented in Table 1 to Table 3 for various time levels and different diffusion coefficient α.

Table 1

Comparison of the numerical solution of u for 1D viscous Burgers equation at different times with $\alpha = 1$

Method	*time*	*x=0.1*	*x=0.3*	*x=0.5*	*x=0.7*	*x=0.9*
IFDM	*0.05*	*0.17832*	*0.47658*	*0.60984*	*0.51165*	*0.20006*
BEM		*0.17759*	*0.47531*	*0.60851*	*0.51050*	*0.19933*
TV-LDG (k = 1)		*0.17807*	*0.47590*	*0.60905*	*0.51108*	*0.19982*
TV-LDG (k = 2)		*0.17803*	*0.47586*	*0.60907*	*0.51112*	*0.19988*
Exact		*0.17803*	*0.47586*	*0.60907*	*0.51113*	*0.19989*
IFDM	*0.1*	*0.11009*	*0.29335*	*0.37342*	*0.31144*	*0.12128*
BEM		*0.10931*	*0.29124*	*0.37070*	*0.30911*	*0.12031*
TV-LDG (k = 1)		*0.10956*	*0.29192*	*0.37162*	*0.31014*	*0.12068*
TV-LDG (k = 2)		*0.10954*	*0.29189*	*0.37158*	*0.30990*	*0.12068*
Exact		*0.10954*	*0.29190*	*0.37158*	*0.30991*	*0.12069*
IFDM	*0.2*	*0.04273*	*0.11276*	*0.14120*	*0.11574*	*0.04457*
BEM		*0.04220*	*0.11044*	*0.13809*	*0.11322*	*0.04391*
TV-LDG (k = 1)		*0.04215*	*0.11121*	*0.13906*	*0.11381*	*0.04380*
TV-LDG (k = 2)		*0.04192*	*0.11062*	*0.13847*	*0.11348*	*0.04368*
Exact		*0.04193*	*0.11062*	*0.13847*	*0.11347*	*0.04369*

Table 2

Comparison of the numerical solution of *u* for 1D viscous Burgers equation at different times with *α= 0.1*

Method	Time	x=0.1	x=0.3	x=0.5	x=0.7	x=0.9
IFDM	0.5	0.11048	0.32367	0.50447	0.57664	0.30912
BEM		0.10986	0.32191	0.50240	0.57514	0.30779
TV-LDG (k = 1)		0.10981	0.32213	0.50277	0.57578	0.30915
TV-LDG (k = 2)		0.10991	0.32219	0.50279	0.57586	0.30932
Exact		0.10992	0.32219	0.50279	0.57585	0.30935
IFDM	1	0.06689	0.19445	0.29448	0.31107	0.14769
BEM		0.06644	0.19263	0.29139	0.30711	0.14507
TV-LDG (k = 1)		0.06629	0.19275	0.29187	0.30799	0.14597
TV-LDG (k = 2)		0.06631	0.19278	0.29192	0.30810	0.14613
Exact		0.06632	0.19279	0.29192	0.30809	0.14607
IFDM	2	0.02909	0.08044	0.10939	0.09838	0.04037
BEM		0.02913	0.07951	0.10770	0.09663	0.03976
TV-LDG (k = 1)		0.02887	0.07973	0.10787	0.09681	0.03966
TV-LDG (k = 2)		0.02876	0.07946	0.10790	0.09688	0.03967
Exact		0.02876	0.07946	0.10789	0.09685	0.03969

Table 3

Comparison of the numerical solution of *u* for 1D viscous Burgers equation at different times with *α= 0.01*

Method	Time	x=0.1	x=0.3	x=0.5	x=0.7	x=0.9
IFDM	0.5	0.12182	0.36206	0.59079	0.79416	0.93322
BEM		0.12079	0.36113	0.59559	0.81257	0.97184
TV-LDG (k = 1)		0.12084	0.36025	0.58869	0.79350	0.93867
TV-LDG (k = 2)		0.12114	0.36027	0.58870	0.79349	0.93809
Exact		0.12114	0.36027	0.58870	0.79349	0.93811
IFDM	2	0.04367	0.13095	0.21800	0.30466	0.38024
BEM		0.043	0.12877	0.21468	0.30075	0.37452
TV-LDG (k = 1)		0.04282	0.12873	0.21450	0.29996	0.37372
TV-LDG (k = 2)		0.04296	0.12884	0.21455	0.29999	0.37325
Exact		0.04296	0.12884	0.21456	0.30000	0.37328
IFDM	4	0.02364	0.07092	0.11817	0.16499	0.17226
BEM		0.02324	0.06935	0.11550	0.16125	0.16515
TV-LDG (k = 1)		0.02310	0.06930	0.11549	0.16122	0.16674
TV-LDG (k = 2)		0.02310	0.06931	0.11549	0.16121	0.16606
Exact		0.02310	0.06931	0.11549	0.16121	0.16606

Fig. 1 to Fig. 4 show the diversities of the numerical solutions with non-identical diffusion coefficients $\alpha = 1, 0.1, 0.01$ *and* 0.005 at different time levels, the main observations that could be noted are that the intensity and the speed of the wave damping are strongly proportional to the value of the diffusion coefficient, in contrast, the shock wave formation is rapidly generated with low values of diffusion coefficient.

From Fig. 3 and Fig. 4 it can be clearly seen that the scheme treats effectively with strong discontinuities without producing nonphysical oscillations, from these features it can be inferred that the TV-LDG method is one of the most efficient methods for solving nonlinear partial differential equations.

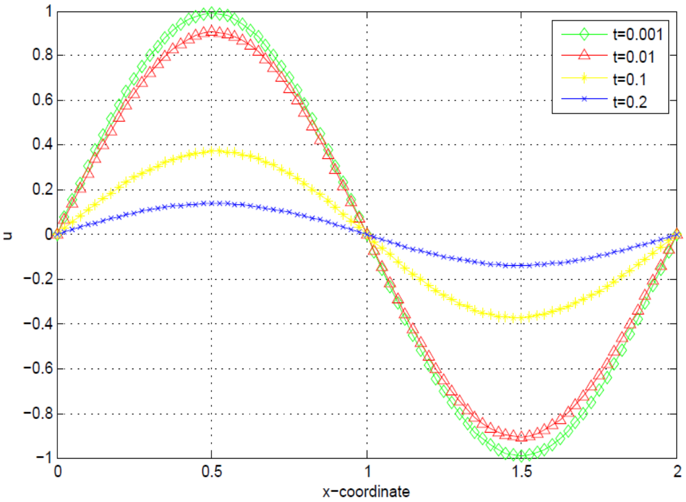

Fig. 1. Numerical solution of at different times for *k=2*, diffusion coefficient *α=1*

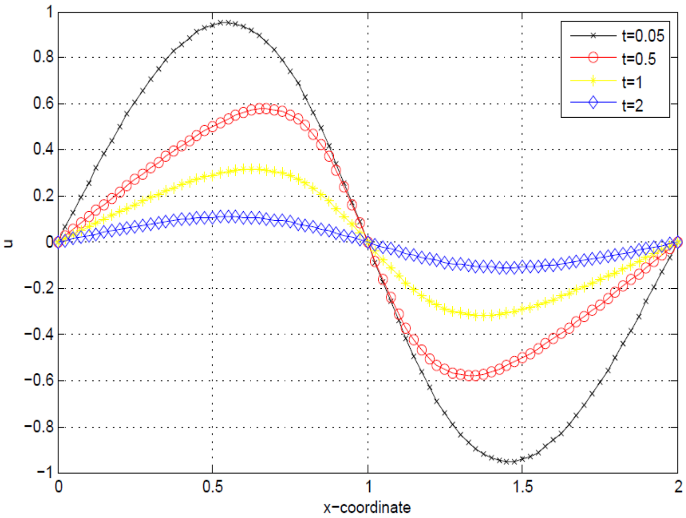

Fig. 2. Numerical solution of at different times for *k=2*, diffusion coefficient *α=0.1*

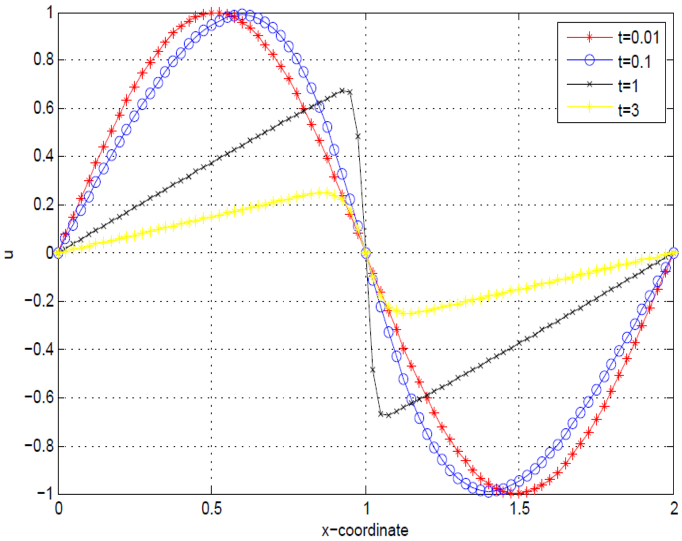

Fig. 3. Numerical solution of at different times for *k=2*, diffusion coefficient *α=0.01*

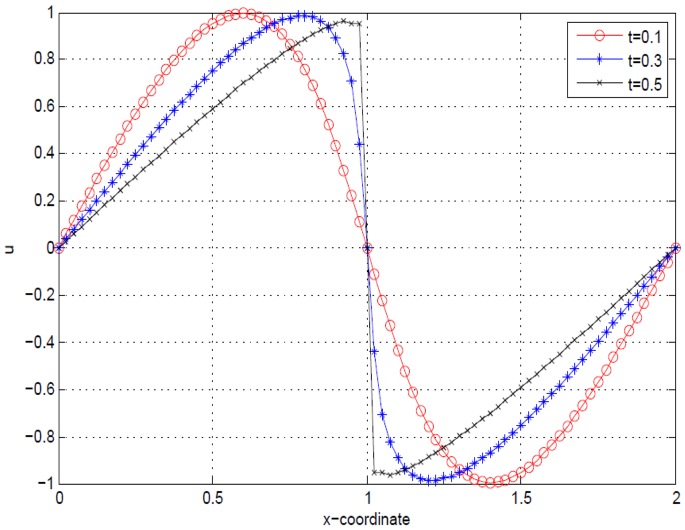

Fig. 4. Numerical solution of at different times for *k=2*, diffusion coefficient *α=0.005*

The numerical simulation at fixed time $t = 0.1$ and different α in Fig. 5 and Fig. 6 are drawn. The tendency of the numerical solutions toward equilibrium and uniformity with the increasing of diffusion coefficient α is shown in Fig. 5 and Fig. 6.

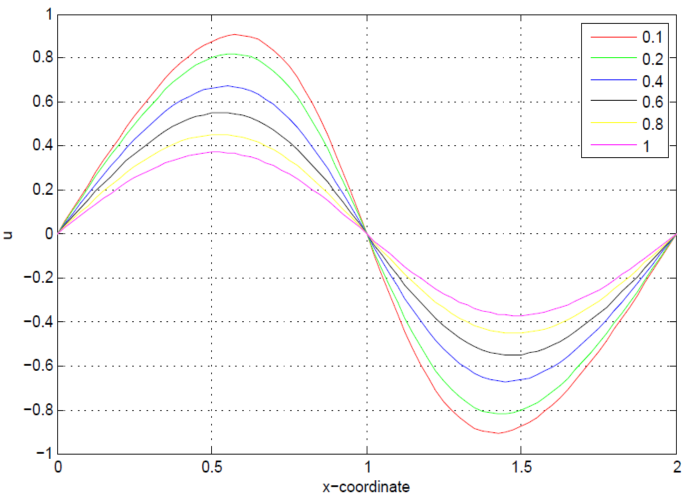

Fig. 5. Numerical solution of at time *t=0.1* for *k=2*, and diffusion coefficients *(α=0.1 to 1)*

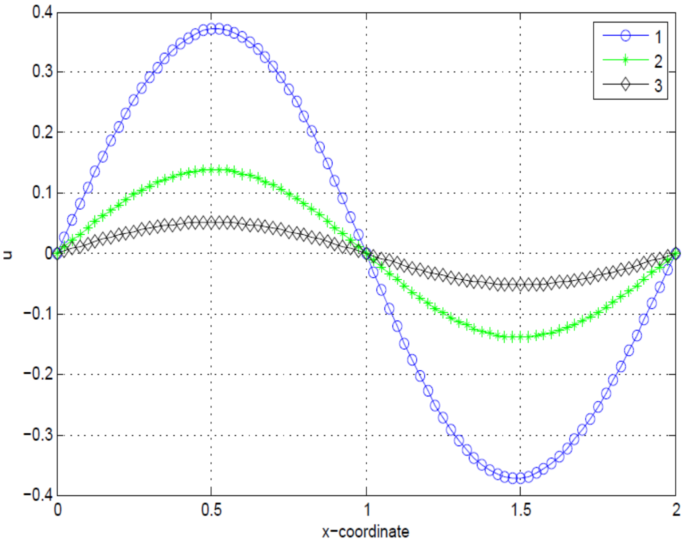

Fig. 6. Numerical solution of at time *t=0.1* for *k=2*, and diffusion coefficients *(α=1 to 3)*

4.2. Test example 2

In this numerical experiment, the system of two-dimensional Burgers' is considered, and the equations given in Eq. (17) and Eq. (18) over a square domain $\Omega = \{(x,y)|0 \le (x,y) \ge 1\}$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{17}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = \alpha\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{18}$$

with the exact solution

$$u(x,y,t) = 0.75 - \frac{0.25}{\left[1 + exp\left\{\frac{4y - 4x - t}{32\alpha}\right\}\right]}$$

$$v(x,y,t) = 0.75 + \frac{0.25}{\left[1 + exp\left\{\frac{4y - 4x - t}{32\alpha}\right\}\right]}$$

The initial and boundary conditions are taken from the exact solutions. The numerical results are computed by using quadratic element *k = 2* and time step length $\Delta t = 10^{-3}$.The numerical and exact values of *u* and *v* at some mesh point for $\alpha = 0.01$ at time levels $t = 0.01,\ 0.5$, and 2.0 are reported in Tables 4-6. The tables show that the present method gives much better results in comparison to methods suggested by Bahadir (2003) and Mittal and Jiwari (2009). To see if the numerical solutions were converging to the exact solution the maximum error L_∞ was computed. For a sufficiently small mesh size $h = \Delta x = \Delta y$, a plot of log(error) versus log(h) will produce a line, the slope of this line is the rate of convergence, which means that the reduction of the error is proportional to the refinement of the mesh size. This is generally referred to as the rate of convergence or order of convergence. The $\log(L_\infty)$ error versus log(h) is depicted in Fig. 7 using the quadratic element *k = 2*. Two main points should be taken away from Fig. 7. First, it can be observed that the error is decreasing as the mesh spacing is decreased. Second, the approximate solution is in fact converging to the exact solution with a proper rate. Fig. 8 reveals the evolution of the computed solution *u* over the time $t = 0,\ 1,$ and 3 using linear element *k=1* and mesh size $\Delta x = \Delta y = 0.1$.

Table 4

Comparison between the exact and the numerical solutions of Example 2, α=0.01 at t = 0.01 with mesh size h=0.05

Mesh point (x,y)	*u*				*v*			
	Bahadir (2003)	*Mittal and Jiwari (2009)*	*Present method*	*Exact solution*	*Bahadir (2003)*	*Mittal and Jiwari (2009)*	*Present method*	*Exact solution*
(0.1,0.1)	*0.62310*	*0.62305*	*0.623041*	*0.623047*	*0.87688*	*0.87695*	*0.876956*	*0.876953*
(0.5,0.1)	*0.50161*	*0.50162*	*0.501622*	*0.501622*	*0.99837*	*0.99838*	*0.998378*	*0.998378*
(0.9,0.1)	*0.50000*	*0.50001*	*0.500011*	*0.500011*	*0.99998*	*0.99999*	*0.999989*	*0.999989*
(0.3,0.3)	*0.62311*	*0.62305*	*0.623041*	*0.623047*	*0.87689*	*0.87695*	*0.876956*	*0.876953*
(0.7,0.3)	*0.50162*	*0.50162*	*0.501622*	*0.501622*	*0.99838*	*0.99838*	*0.998378*	*0.998378*
(0.1,0.5)	*0.74827*	*0.74827*	*0.748275*	*0.748274*	*0.75172*	*0.75172*	*0.751725*	*0.751726*
(0.5,0.5)	*0.62311*	*0.62305*	*0.623041*	*0.623047*	*0.87689*	*0.87695*	*0.876956*	*0.876953*
(0.9,0.5)	*0.50162*	*0.50162*	*0.501622*	*0.501622*	*0.99838*	*0.99838*	*0.998378*	*0.998378*
(0.3,0.7)	*0.74827*	*0.74827*	*0.748275*	*0.748274*	*0.75173*	*0.75173*	*0.751725*	*0.751726*
(0.7,0.7)	*0.62311*	*0.62305*	*0.623041*	*0.623047*	*0.87689*	*0.87695*	*0.876956*	*0.876953*
(0.1,0.9)	*0.74998*	*0.74999*	*0.749988*	*0.749988*	*0.75001*	*0.75001*	*0.750012*	*0.750022*
(0.5,0.9)	*0.74827*	*0.74827*	*0.748275*	*0.748274*	*0.75173*	*0.75172*	*0.751725*	*0.751726*
(0.9,0.9)	*0.62311*	*0.62305*	*0.623041*	*0.623047*	*0.87689*	*0.87695*	*0.876956*	*0.876953*

Table 5

Comparison between the exact and the numerical solutions of Example 2, α=0.01 at t = 0.5 with mesh size h=0.05

Mesh point (x,y)	*u*					*v*			
	Bahadir (2003)	Mittal and Jiwari (2009)	Present method	Exact solution		Bahadir (2003)	Mittal and Jiwari (2009)	Present method	Exact Solution
(0.1,0.1)	0.54235	0.54322	0.543323	0.543322		0.95577	0.95678	0.956679	0.956678
(0.5,0.1)	0.49964	0.50035	0.500351	0.500353		0.99827	0.99965	0.999648	0.999647
(0.9,0.1)	0.49931	0.50000	0.500003	0.500002		0.99861	1.00000	0.999998	0.999998
(0.3,0.3)	0.54207	0.54321	0.543327	0.543322		0.95596	0.95679	0.956677	0.956678
(0.7,0.3)	0.49961	0.50035	0.500352	0.500353		0.99827	0.99964	0.999647	0.999647
(0.1,0.5)	0.74130	0.74219	0.742215	0.742214		0.75699	0.75780	0.757786	0.757786
(0.5,0.5)	0.54222	0.54329	0.543322	0.543322		0.95685	0.95671	0.956679	0.956678
(0.9,0.5)	0.49997	0.50035	0.500353	0.500353		0.99903	0.99965	0.999647	0.999647
(0.3,0.7)	0.74146	0.74221	0.742216	0.742214		0.75723	0.75779	0.757784	0.757786
(0.7,0.7)	0.54243	0.54332	0.543321	0.543322		0.95746	0.95668	0.956680	0.956678
(0.1,0.9)	0.74913	0.74995	0.749946	0.749946		0.74924	0.75005	0.750054	0.750054
(0.5,0.9)	0.74201	0.74221	0.742216	0.742214		0.75781	0.75779	0.757785	0.757786
(0.9,0.9)	0.54232	0.54332	0.543325	0.543322		0.95777	0.95667	0.956678	0.956678

Table 6

Comparison between the exact and the numerical solutions of Example 2, α=0.01 at t = 2 with mesh size h=0.05

Mesh point (x,y)	*u*					*v*			
	Bahadir (2003)	Mittal and Jiwari (2009)	Present method	Exact solution		Bahadir (2003)	Mittal and Jiwari (2009)	Present method	Exact solution
(0.1,0.1)	0.49983	0.50048	0.500487	0.500482		0.99826	0.99952	0.999516	0.999518
(0.5,0.1)	0.49930	0.50000	0.500002	0.500003		0.99860	1.00000	0.999997	0.999997
(0.9,0.1)	0.49930	0.50000	0.500002	0.500000		0.99861	1.00000	0.999999	0.999999
(0.3,0.3)	0.49977	0.50048	0.500490	0.500482		0.99820	0.99952	0.999514	0.999518
(0.7,0.3)	0.49930	0.50000	0.500002	0.500003		0.99860	1.00000	0.999997	0.999997
(0.1,0.5)	0.55461	0.55540	0.555647	0.555675		0.94393	0.94460	0.944349	0.944325
(0.5,0.5)	0.49973	0.50048	0.500484	0.500482		0.99821	0.99952	0.999517	0.999518
(0.9,0.5)	0.49931	0.50000	0.500003	0.500003		0.99862	1.00000	0.999997	0.999997
(0.3,0.7)	0.55429	0.55540	0.555675	0.555675		0.94409	0.94460	0.944331	0.944325
(0.7,0.7)	0.49970	0.50048	0.500480	0.500482		0.99823	0.99952	0.999519	0.999518
(0.1,0.9)	0.74340	0.74422	0.744256	0.744256		0.75500	0.75578	0.755744	0.755744
(0.5,0.9)	0.55413	0.55541	0.555671	0.555675		0.94441	0.94459	0.944332	0.944325
(0.9,0.9)	0.50001	0.50048	0.500475	0.500482		0.99846	0.99952	0.999522	0.999518

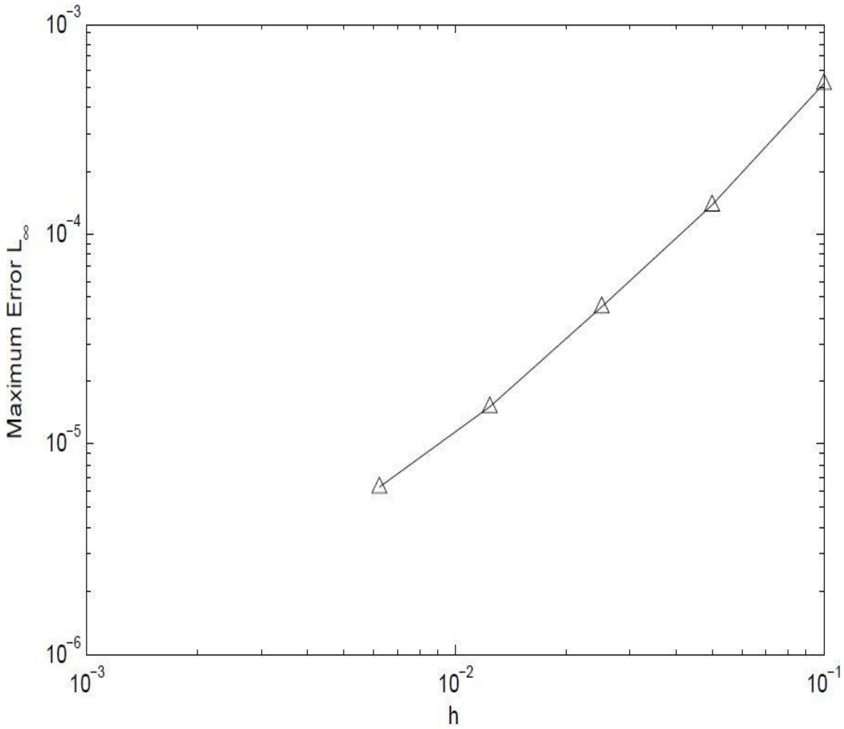

Fig. 7. The rate of convergence by using quadratic element, with different mesh size ($\Delta x = \Delta y = \frac{1}{10}, \frac{1}{20}, \frac{1}{40}, \frac{1}{80}$ **and** $\frac{1}{160}$)

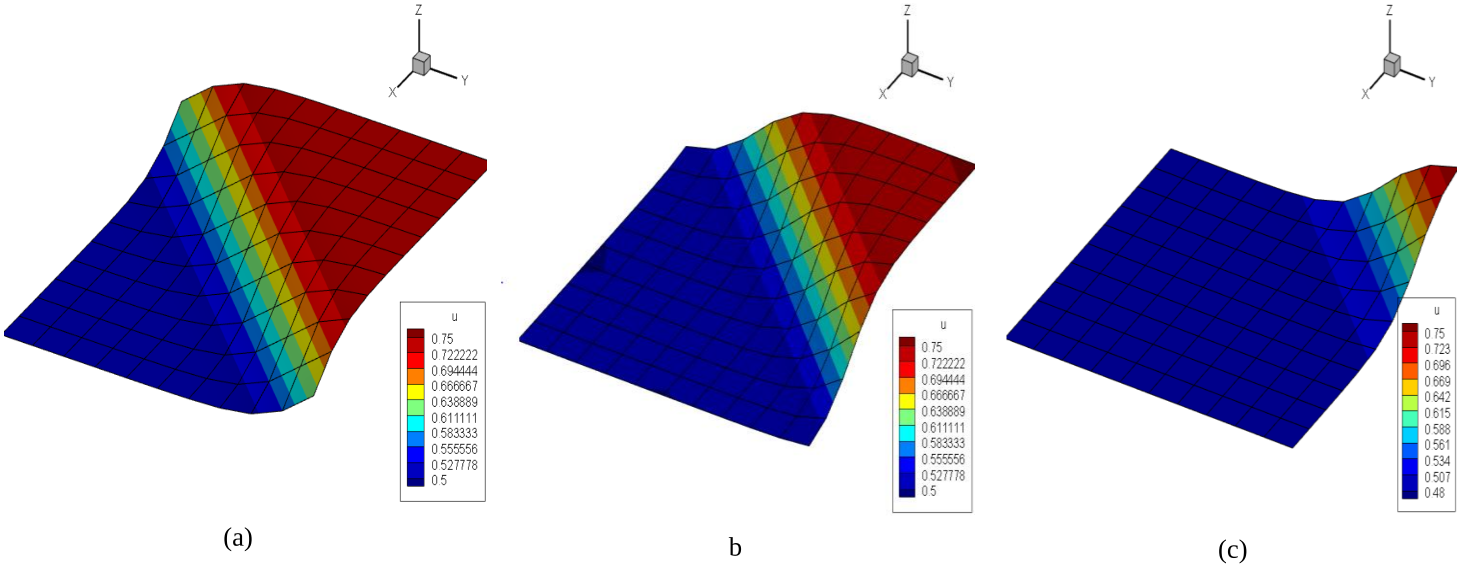

Fig. 8. The evolution of the numerical solution for u over time. (a) at t=0; (b) at t=1 and (c) at t=3. with mesh size $\Delta x = \Delta y = \frac{1}{10}$

5. Conclusions

Constructing high-order accurate totally, volume local discontinuous Galerkin finite element method for the numerical solution of the viscous Burgers' equations on Cartesian meshes has been made successfully. Based on the results of the present method, the following observations and inferences can be drawn:

- The present method reveals a good capturing of discontinuity when it deals with the shocking flow, which makes the TV-LDG as a reliable method for solving more general problems in fluid dynamics with a low diffusion coefficient.
- The proposed scheme has a unique agreement with the analytical solution and gives lower error in comparison with the other existing numerical solutions, in the future, the main target is that extend this study by applying the scheme on more complex partial differential equations.

References

Abdou, M. Soliman, A. (2005) 'Variational iteration method for solving Burger's and coupled Burger's equations', *Journal of Computational and Applied,* 181, pp. 245-251.

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

Bahadir, A. (1999) 'Numerical solution for one-dimensional Burgers' equation using a fully implicit finite-difference method', *International Journal of Applied Mathematics*, 8, pp. 897–909.

Bahadir, A. and Mustafa, S. (2005) 'Mixed finite difference and boundary element approach to one-dimensional Burgers' equation', *Applied Mathematics and Computation,* 160, pp. 663–673.

Bahadir, A. (2003) 'Fully implicit finite-difference scheme for two-dimensional Burgers equations', *Applied Mathematics Computation*, 137, pp. 131–137.

Bassi, F. and Rebay, S. (1997) 'A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations', *Journal of Computational Physics*, 131, pp. 267-279.

Burgers, J.M. (1948) 'A mathematical model illustrating the theory of turbulence', Advances in Applied Mechanics, 1, pp. 171-199.

Cockburn, B. and Shu, C. (1998) 'The local discontinuous Galerkin method for time-dependent convection-diffusion systems, *SIAM Journal on Numerical Analysis,* 35, pp. 2440–2463.

Elhadi, E., Fakroon, M. and Abdelhadi, A. (2020) 'Solving the unsteady linear advection-diffusion equations by using the totally volume integral of the local discontinuous Galerkin method', *Libyan Journal of Science and Technology,* 11, pp. 53-60

Mittal, R. and Jiwari, R. (2009)'Differential quadrature method for two-dimensional Burgers'equations', *International Journal for Computational Methods in Engineering Science and Mechanics,* 10, pp. 450-459.

Shu, C. and Osher, S. (1988) 'Efficient implementation of essentially nonoscillatory shock-capturing schemes'. *Journal of Computational Physics*, 77, pp. 439–471.

Soliman, A. (2006) 'The modified extended tanh-function method for solving Burgers-type equations', *Physica A: Statistical Mechanics and its Applications*, 361, pp. 394-404.

Taku, O. (2009) 'Cole–Hope transformation as a numerical tool for the Burgers equation', *Applied Mathematics and Computation*, 8, pp 107–113.

Toro, E. F. (1999) *Riemann solvers and numerical methods for fluid dynamics.* 2nd edn. Springer.