

A STABILIZED DISCONTINUOUS GALERKIN METHOD FOR THE NONLINEAR ADVECTION-DIFFUSION PROCESSES

HUSEYIN TUNC AND MURAT SARI

Abstract. This article presents a hybridization of a local discontinuous Galerkin method (LDG) with the θ -method to capture nonlinear behavior of the advection-diffusion processes. The predetermined fixed flux selection is extended to the generalized problem-dependent flux selection in the LDG algorithm. The derived technique has been shown to be unconditionally stable through the L^2 stability analysis. Two illustrative test problems are considered to demonstrate the efficiency of the currently produced technique for both advection and diffusion dominated processes.

1. Introduction

In this paper, we consider the following Burgers equation,

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \epsilon \frac{\partial^2 u}{\partial x^2}, \quad x \in (0, 1), \quad t \in (0, T] \quad (1.1)$$

with initial condition

$$u(x, 0) = f(x), \quad x \in (0, 1) \quad (1.2)$$

and boundary conditions

$$u(0, t) = 0 \quad \text{and} \quad u(1, t) = 0 \quad (1.3)$$

where ϵ is the kinematic viscosity constant for $\epsilon > 0$ and f is a sufficiently smooth function. The Burgers equation is the nonlinear model equation for diffusive waves in fluid dynamics. The corresponding equation has also many application areas including theory of shock waves, sound waves in a viscous medium, mathematical modeling of turbulent fluid and so on. The Burgers equation firstly introduced by Bateman [5] and proposed the steady-state solution of the equation. Burgers [6] introduced this equation to model turbulent fluid in a channel caused by the opposite effects of convection and diffusion. The structure of the Burgers equation is similar to that of the Navier-Stokes equations without stress term due to the presence of the nonlinear convection term and the occurrence of the diffusion term with viscosity coefficient. The Burgers equation was solved exactly by using the Hopf-Cole transformation [13, 9] which converts the equation

2020 *Mathematics Subject Classification.* 65M60, 65N30, 35L67.

Key words and phrases. Advection-diffusion processes, Local discontinuous Galerkin method, Burgers equation, θ -method, Hopf-Cole transformation.

to a heat diffusion equation. In most of those cases, the solutions involve infinite series which may diverge or converge very slowly for relatively small values of the kinematic viscosity constant ϵ , which corresponds to steep wave fronts in the propagation of the dynamic wave forms.

So far various numerical methods such as finite element based methods [23, 24, 17, 27], finite difference based methods [26, 25], a hybrid numerical scheme including wavelets and finite differences [14], sinc based differential quadrature method [16], spline-based finite difference method [11], automatic differentiation method [3], boundary element method [4], Haar wavelet quasi-linearization approach [15] have been developed in dealing with numerical solutions of the Burgers equation.

In recent decades, the local discontinuous Galerkin (LDG) method has gained popularity for solving partial differential equations due to their computational flexibility and ability to incorporate physical properties [27, 21, 7, 8, 22, 2, 28]. The discontinuous Galerkin (DG) method was first introduced by Reed and Hill [21] for the solution of the steady state neutron transport equation. Since then, the DG methods have been developed rapidly because of the several advantages such as high geometric flexibility, and mass conservation properties. In 1998 Cockburn and Shu [7] introduced the LDG method for solving the nonlinear convection diffusion equation. This technique was in turn an extension of the Runge-Kutta discontinuous Galerkin method developed by Cockburn et al. [8] for nonlinear hyperbolic systems. Pei et al. [22] solved the Burgers equation by using the LDG in space discretization and the TVD Runge-Kutta method with local Lax-Friedrichs flux. In the study of Shao et al. [27], the Burgers equation was transformed to the linear diffusion equation and then solved by using the Legendre polynomials based LDG method with the Runge-Kutta method.

In the present study we prefer to transform the Burgers equation to a linear diffusion equation and the transformed equation is solved by the LDG method with general flux selection. The resulted ODE system is then solved by the θ -method to preserve the unconditional stability of the fully discretized system. The general parameter-based flux selection is derived and implemented in the LDG formulation to get adaptive flux selection for challenging stiff cases. It has been proved with the L^2 stability analysis that the current LDG method satisfies stability requirements irrespective of the flux parameter values. Two challenging test problems are considered, qualitative and quantitative results are presented in the text. The produced results have been seen to be unconditionally stable and highly accurate. The efficiency of the present method is presented in terms of the changing values of local polynomial degrees, local flux-parameters and kinematic viscosity constants.

2. Local discontinuous Galerkin method

2.1. Hopf-Cole transformation. With the use of Hopf-Cole transformation [13, 9], the Burgers equation (1.1) is transformed into a diffusion equation as follows,

$$u(x, t) = -2\epsilon \frac{c_x(x, t)}{c(x, t)}. \quad (2.1)$$

After transforming the Burgers equation, $c(x, t)$ satisfies the following linear diffusion equation

$$\frac{\partial c}{\partial t} = \epsilon \frac{\partial^2 c}{\partial x^2}, \quad x \in (0, 1), \quad t \in (0, T]. \quad (2.2)$$

with initial condition

$$c(x, 0) = \exp \left\{ - \int_0^x \frac{f(s)}{2\epsilon} ds \right\}, \quad x \in (0, 1) \quad (2.3)$$

and homogeneous Neumann boundary conditions

$$c_x(0, t) = 0 \quad \text{and} \quad c_x(1, t) = 0. \quad (2.4)$$

2.2. The LDG formulation. To construct the LDG method for equation (2.2) with conditions (2.3)-(2.4), under the consideration of variable $p = \sqrt{\epsilon}c_x$, the diffusion equation can be rewritten in the following hyperbolic form

$$\begin{aligned} c_t - \sqrt{\epsilon}p_x &= 0 \\ p - \sqrt{\epsilon}c_x &= 0 \end{aligned} \quad (2.5)$$

with the Neumann boundary conditions (2.4) for $c(x, t)$, homogenous Dirichlet boundary conditions for $p(x, t)$ and initial condition (2.3). To define finite elements of the spatial domain, the interval $[0, 1]$ is partitioned into K elements. The elements are the member of mesh $\omega = \{I_k = (x_k, x_{k+1}), k = 0, 1, \dots, K-1\}$ with the center of element $x_{k+\frac{1}{2}} = (x_k + x_{k+1})/2$ where $0 = x_0 < x_1 < \dots < x_K = 1$ and the element length is fixed and defined as $h = 1/K$. For the mesh ω , we introduce the broken Sobolev space

$$H^1((0, 1), \omega) = \{v : (0, 1) \rightarrow R \mid v|_{I_j} \in H^1(I_j), j = 0, 1, \dots, K-1\} \quad (2.6)$$

which allows us to define discontinuous solutions. We use the notation $c_j^\pm = \lim_{x \rightarrow x_j^\pm} c(x, t)$ over the interface points. Assume that the functions $\hat{c}_h(x, t)$ and $\hat{p}_h(x, t)$ of the space $H^1(0, T : V_h^N)$ are piecewise discontinuous approximate functions of $c(x, t)$ and $p(x, t)$ where finite dimensional subspace V_h^N is defined as

$$V_h^N = \{u : (0, 1) \rightarrow R \mid u|_{I_j} \in P_N, j = 0, 1, \dots, K-1\} \subset H^1((0, 1), \omega). \quad (2.7)$$

The approximate local solutions can be stated as in the following nodal form

$$\begin{aligned} \hat{c}_h^k(x, t) &= \sum_{n=1}^{N_p} \beta_n^k \varphi_n(x), \quad x \in [x_k^+, x_{k+1}^-] \\ \hat{p}_h^k(x, t) &= \sum_{n=1}^{N_p} \delta_n^k \varphi_n(x), \quad x \in [x_k^+, x_{k+1}^-] \end{aligned} \quad (2.8)$$

where β_n^k and δ_n^k are time dependent quantities, $\varphi_n(x)$ are the Lagrange polynomials and $N = N_p - 1$ is the degree of the polynomial approximation. The interpolation nodes of the element $I_k = (x_k, x_{k+1})$ are selected as the Gauss-Lobatto grid points to minimize the approximation error and to use non-oscillatory high degree polynomials. By doing hp -refinement, the error can be minimized either by decreasing h or by increasing polynomial degree. More details on the basis

functions can be found in the literature [12]. To find out weak formulation of the system (2.5), let us multiply both equations by test functions $v_h, z_h \in V_h^N$, then integrate over the element $I_k = (x_k, x_{k+1})$ and the use of the integration by parts yields

$$\begin{aligned} \int_{I_k} (\hat{c}_h^k)_t v_h^k dx + \sqrt{\epsilon} \int_{I_k} \hat{p}_h^k (v_h^k)_x dx + p^* v_h^k \Big|_{x_k^+}^{x_{k+1}^-} &= 0 \\ \int_{I_k} \hat{p}_h^k z_h^k dx + \sqrt{\epsilon} \int_{I_k} \hat{c}_h^k (z_h^k)_x dx + c^* z_h^k \Big|_{x_k^+}^{x_{k+1}^-} &= 0 \end{aligned} \quad (2.9)$$

where p^* and c^* are numerical flux functions which are crucial for the stability as well as the accuracy. In this study, we consider the following selections

$$c^*(x_k, t) = \sqrt{\epsilon} \begin{cases} (\hat{c}_h)_0^+, & k = 0 \\ \alpha (\hat{c}_h)_k^- + (1 - \alpha) (\hat{c}_h)_k^+, & k = 1, \dots, K - 1 \\ (\hat{c}_h)_K^-, & k = K \end{cases} \quad (2.10)$$

$$p^*(x_k, t) = \sqrt{\epsilon} \begin{cases} (\hat{p}_h)_0^+, & k = 0 \\ (1 - \alpha) (\hat{p}_h)_k^- + \alpha (\hat{p}_h)_k^+, & k = 1, \dots, K - 1 \\ (\hat{p}_h)_K^-, & k = K \end{cases} \quad (2.11)$$

where $c_h(x, t)$ and $p_h(x, t)$ stand for global approximate solutions. Numerical flux parameter determines the values of the unknown solutions over the interface points with $0 \leq \alpha \leq 1$. To be able to use the Gauss-Lobatto grid points, we convert the interval $I_k = (x_k, x_{k+1})$ to the interval $I = (-1, 1)$ by using the following transformation

$$x(r) = x_k + \frac{1+r}{2}h \quad (2.12)$$

with the reference variable $r \in I = (-1, 1)$ for all $k = 0, 1, \dots, K - 1$. Writing the approximate solutions (2.8) in system (2.9), using the flux selections (2.10)-(2.11) and considering the test functions as basis functions lead to the following matrix system for element $I_k = (x_k, x_{k+1})$

$$\begin{aligned} A^k \frac{d\beta^k}{dt} + \sqrt{\epsilon} B^k \delta^k &= \sqrt{\epsilon} C^k \delta^k \\ A^k \delta^k + \sqrt{\epsilon} B^k \beta^k &= \sqrt{\epsilon} D^k \beta^k \end{aligned} \quad (2.13)$$

where

$$A_{ij} = \frac{h}{2} \int_I \varphi_i(r) \varphi_j(r) dr, \quad (2.14)$$

$$B_{ij} = \int_I \frac{d\varphi_i(r)}{dx} \varphi_j(r) dr, \quad (2.15)$$

$$s_1 \delta_{j_1} + s_2 \delta_{j_2} = p^* \varphi_i(r) \Big|_{-1}^1 \rightarrow C_{ij_1}^k = s_1 \quad \text{and} \quad C_{ij_2}^k = s_2, \quad (2.16)$$

$$z_1 \beta_{j_1} + z_2 \beta_{j_2} = c^* \varphi_i(r) \Big|_{-1}^1 \rightarrow D_{ij_1}^k = z_1 \quad \text{and} \quad D_{ij_2}^k = z_2, \quad (2.17)$$

$$\beta^k = [\beta_1^k, \beta_2^k, \dots, \beta_{N_p}^k]^T, \quad (2.18)$$

$$\delta^k = [\delta_1^k, \delta_2^k, \dots, \delta_{N_p}^k]^T, \quad (2.19)$$

for $i, j = 1, 2, \dots, N_p$ and $k = 0, 1, \dots, K - 1$. Assembling each element and imposing the boundary conditions lead to the following global system

$$\begin{aligned} A_c \frac{d\beta}{dt} + \sqrt{\epsilon} B_p \delta &= \sqrt{\epsilon} C \delta \\ A_p \delta + \sqrt{\epsilon} B_c \beta &= \sqrt{\epsilon} D \beta \end{aligned} \quad (2.20)$$

where the matrices are $(N_p K - 2) \times (N_p K - 2)$ and independent of time. Note that δ can be calculated at each step directly from (2.20) and writing it in the first part yields

$$A_c \frac{d\beta}{dt} + \epsilon (B_p - C) [A_p^{-1} (D - B_c)] \beta = 0. \quad (2.21)$$

Equation (2.21) is an ODE system with vector $\beta = [\beta_2^0, \beta_3^0, \dots, \beta_{N_p-1}^{K-1}]^T$ and will be solved by using the θ -method in the following subsection.

2.3. θ -family of the time approximation. From our final matrix form (2.21), one can write

$$A_c \frac{d\beta}{dt} + R \beta = 0 \quad (2.22)$$

where $R = \epsilon (B_p - C) [A_p^{-1} (D - B_c)]$. As stated in [29] the θ -family of the approximation can be defined as

$$\{\beta\}_{s+1} = \{\beta\}_s + dt \{\beta\}_{s+\theta} \quad (2.23)$$

$$\{\beta\}_{s+\theta} = (1 - \theta) \left\{ \dot{\beta} \right\}_s + \theta \left\{ \dot{\beta} \right\}_{s+1} \quad (2.24)$$

or

$$dt \left[(1 - \theta) \left\{ \dot{\beta} \right\}_s + \theta \left\{ \dot{\beta} \right\}_{s+1} \right] = \{\beta\}_{s+1} - \{\beta\}_s \quad (2.25)$$

where $0 \leq \theta \leq 1$ and $t_{s+1} - t_s = dt$. Here $\dot{\beta}$ stands for the time differentiation. Following the procedure given in [29], (2.22) becomes

$$[A_c + \theta dt R] \{\beta\}_{s+1} = [A_c - (1 - \theta) dt R] \{\beta\}_s. \quad (2.26)$$

Equation (2.26) is a recursive relation of $\{\beta\}_s$. By obtaining $\{\beta\}_0$ from (2.3), the rest of the unknown vectors can be evaluated.

3. Stability Analysis

In the present section, the stability analysis of the semi-discrete LDG formulation has been carried out by the energy method. It is important to obtain a stable spatial discretization to make sense with physical observations of the problem. However, to analyze physical system properly, it is also necessary to consider temporal variation of the process. At this moment, the selection of the time integration method plays an important role, i.e. the fully discrete hybrid scheme (2.26) must be stable in time. The θ -family of the time approximation leads to unconditionally stable fully discrete scheme ($\theta \geq 0.5$) under the consideration of stable spatial approximation [29].

In literature [27], the researchers proposed a stability analysis for the LDG approach in terms of the energy method. We have expanded this idea to general flux selections (2.10)-(2.11) from their specific selection, $\alpha = 0$. Then we have

proved that for all selections of the flux parameter $\alpha \in [0, 1]$, one can readily obtain the “cell entropy inequality” as is the case in the literature [27].

Proposition 3.1. *The approximate solutions \hat{c}_h and \hat{p}_h for semi-discrete LDG formulation (2.9) satisfy the following “cell entropy inequality”*

$$\frac{1}{2} \frac{d}{dt} \int_{I_k} (\hat{c}_h)^2 dx + \int_{I_k} (\hat{p}_h)^2 dx + \hat{H}_{k+1} - \hat{H}_k = 0 \quad (3.1)$$

for some consistent entropy flux and

$$\hat{H}_k = \hat{H}(\hat{c}_h(x_k^-, t), \hat{p}_h(x_k^-, t); \hat{c}_h(x_k^+, t), \hat{p}_h(x_k^+, t)) \quad (3.2)$$

satisfying $\hat{H}(\hat{c}_h, \hat{p}_h; \hat{c}_h, \hat{p}_h) = -\sqrt{\epsilon} \hat{c}_h \hat{p}_h$.

Proof. By using equation (2.9) with global approximate solutions, the following functional can be defined,

$$\begin{aligned} B_k(\hat{c}_h, \hat{p}_h; \hat{c}_h, \hat{p}_h) &= \int_{I_k} (\hat{c}_h)_t v_h dx + \sqrt{\epsilon} \int_{I_k} \hat{p}_h (v_h)_x dx - (p^*)_{k+1} (v_h)_{k+1}^- \\ &\quad + (p^*)_k (v_h)_k^+ + \int_{I_k} \hat{p}_h z_h dx + \sqrt{\epsilon} \int_{I_k} \hat{c}_h (z_h)_x dx \\ &\quad - (c^*)_{k+1} (z_h)_{k+1}^- + (c^*)_k (z_h)_k^+. \end{aligned} \quad (3.3)$$

Taking $v_h = \hat{c}_h$ and $z_h = \hat{p}_h$ in the functional leads to

$$\begin{aligned} B_k(\hat{c}_h, \hat{p}_h; \hat{c}_h, \hat{p}_h) &= \frac{1}{2} \frac{d}{dt} \int_{I_k} (\hat{c}_h)^2 dx + \int_{I_k} (\hat{p}_h)^2 dx + \sqrt{\epsilon} \int_{I_k} \hat{c}_h (p_h)_x dx \\ &\quad + \sqrt{\epsilon} \int_{I_k} \hat{p}_h (c_h)_x dx + (p^*)_k (\hat{c}_h)_k^+ + (c^*)_k (\hat{p}_h)_k^+ \\ &\quad - (p^*)_{k+1} (\hat{c}_h)_{k+1}^- - (c^*)_{k+1} (\hat{p}_h)_{k+1}^-. \end{aligned} \quad (3.4)$$

Using the following equality

$$\sqrt{\epsilon} \int_{I_k} \hat{c}_h (p_h)_x dx + \sqrt{\epsilon} \int_{I_k} \hat{p}_h (c_h)_x dx = \sqrt{\epsilon} (\hat{c}_h)_{k+1}^- (\hat{p}_h)_{k+1}^- - \sqrt{\epsilon} (\hat{c}_h)_k^+ (\hat{p}_h)_k^+ \quad (3.5)$$

the functional $B_k(\hat{c}_h, \hat{p}_h; \hat{c}_h, \hat{p}_h)$ becomes:

$$\begin{aligned} B_k(\hat{c}_h, \hat{p}_h; \hat{c}_h, \hat{p}_h) &= \frac{1}{2} \frac{d}{dt} \int_{I_k} (\hat{c}_h)^2 dx + \int_{I_k} (\hat{p}_h)^2 dx + \sqrt{\epsilon} (\hat{c}_h)_{k+1}^- (\hat{p}_h)_{k+1}^- \\ &\quad - (p^*)_{k+1} (\hat{c}_h)_{k+1}^- - (c^*)_{k+1} (\hat{p}_h)_{k+1}^- - \sqrt{\epsilon} (\hat{c}_h)_k^- (\hat{p}_h)_k^- \\ &\quad + (p^*)_k (\hat{c}_h)_k^- + (c^*)_k (\hat{p}_h)_k^- + \sqrt{\epsilon} (\hat{c}_h)_k^- (\hat{p}_h)_k^- \\ &\quad - (p^*)_k (\hat{c}_h)_k^- - (c^*)_k (\hat{p}_h)_k^- - \sqrt{\epsilon} (\hat{c}_h)_k^+ (\hat{p}_h)_k^+ \\ &\quad + (p^*)_k (\hat{c}_h)_k^+ + (c^*)_k (\hat{p}_h)_k^+. \end{aligned} \quad (3.6)$$

Defining new functionals \hat{H} and Θ as

$$\hat{H} = \sqrt{\epsilon} (\hat{c}_h)^- (\hat{p}_h)^- - p^* (\hat{c}_h)^- - c^* (\hat{p}_h)^- \quad (3.7)$$

$$\Theta = \sqrt{\epsilon} (\hat{c}_h)^- (\hat{p}_h)^- - p^* (\hat{c}_h)^- - c^* (\hat{p}_h)^- - \sqrt{\epsilon} (\hat{c}_h)^+ (\hat{p}_h)^+ + p^* (\hat{c}_h)^+ + c^* (\hat{p}_h)^+ \quad (3.8)$$

equation (3.6) can be written as

$$B_k(\hat{c}_h, \hat{p}_h; \hat{c}_h, \hat{p}_h) = \frac{1}{2} \frac{d}{dt} \int_{I_k} (\hat{c}_h)^2 dx + \int_{I_k} (\hat{p}_h)^2 dx + \hat{H}_{k+1} - \hat{H}_k + \Theta_k = 0 \quad (3.9)$$

where $k = 0, 1, \dots, K-1$. By using the flux definitions given in equations (2.10)-(2.11), the functional Θ becomes for interior points

$$\begin{aligned} \Theta_k &= \sqrt{\epsilon} (\hat{c}_h)_k^- (\hat{p}_h)_k^- - \sqrt{\epsilon} ((1-\alpha) (\hat{p}_h)_k^- + \alpha (\hat{p}_h)_k^+) (\hat{c}_h)_k^- \\ &\quad - \sqrt{\epsilon} (\alpha (\hat{c}_h)_k^- + (1-\alpha) (\hat{c}_h)_k^+) (\hat{p}_h)_k^- - \sqrt{\epsilon} (\hat{c}_h)_k^+ (\hat{p}_h)_k^+ \\ &\quad + \sqrt{\epsilon} ((1-\alpha) (\hat{p}_h)_k^- + \alpha (\hat{p}_h)_k^+) (\hat{c}_h)_k^+ \\ &\quad + \sqrt{\epsilon} (\alpha (\hat{c}_h)_k^- + (1-\alpha) (\hat{c}_h)_k^+) (\hat{p}_h)_k^+ \\ &= 0 \end{aligned} \quad (3.10)$$

where $k = 1, \dots, K-1$ and for the left boundary point

$$\begin{aligned} \Theta_0 &= \sqrt{\epsilon} (\hat{c}_h)_0^- (\hat{p}_h)_0^- - \sqrt{\epsilon} (\hat{p}_h)_0^+ (\hat{c}_h)_0^- - \sqrt{\epsilon} (\hat{c}_h)_0^+ (\hat{c}_h)_0^+ (\hat{p}_h)_0^- \\ &\quad - \sqrt{\epsilon} (\hat{c}_h)_0^+ (\hat{p}_h)_0^+ + \sqrt{\epsilon} (\hat{p}_h)_0^+ (\hat{c}_h)_0^+ + \sqrt{\epsilon} (\hat{c}_h)_0^+ (\hat{p}_h)_0^- \\ &= \sqrt{\epsilon} (\hat{c}_h)_0^- ((\hat{p}_h)_0^- - (\hat{p}_h)_0^+) \\ &= 0 \end{aligned} \quad (3.11)$$

Then the proof of Proposition 1 is completed. This proof gives a generalization of the study [27] by considering the parameter-dependent fluxes over the boundaries. \square

Proposition 3.2. *The semi-discrete LDG procedure (2.9) with the Neumann boundary conditions (2.4) satisfies the L^2 -stability*

$$\frac{d}{dt} \int_{I_k} (\hat{c}_h)^2 dx + 2 \int_{I_k} (\hat{p}_h)^2 dx = 0. \quad (3.12)$$

Proof. Summing over all k in the cell entropy inequality yields

$$\frac{1}{2} \frac{d}{dt} \int_{I_k} (\hat{c}_h)^2 dx + \int_{I_k} (\hat{p}_h)^2 dx + \hat{H}_K - \hat{H}_0 = 0 \quad (3.13)$$

in which the residual functional values with the definitions of the flux functions in (2.10)-(2.11) can be stated as

$$\begin{aligned} \hat{H}_K &= \sqrt{\epsilon} (\hat{c}_h)_K^- (\hat{p}_h)_K^- - \sqrt{\epsilon} (\hat{p}_h)_K^- (\hat{c}_h)_K^- - \sqrt{\epsilon} (\hat{c}_h)_K^- (\hat{p}_h)_K^- \\ &= -\sqrt{\epsilon} (\hat{p}_h)_K^- (\hat{c}_h)_K^- \\ &= 0 \end{aligned} \quad (3.14)$$

and

$$\begin{aligned} \hat{H}_0 &= \sqrt{\epsilon} (\hat{c}_h)_0^- (\hat{p}_h)_0^- - \sqrt{\epsilon} (\hat{p}_h)_0^+ (\hat{c}_h)_0^- - \sqrt{\epsilon} (\hat{c}_h)_0^+ (\hat{p}_h)_0^- \\ &= (\hat{c}_h)_0^- ((\hat{p}_h)_0^- - (\hat{p}_h)_0^+) \\ &= 0 \end{aligned} \quad (3.15)$$

where the following Neumann boundary conditions and the flux conditions

$$(\hat{p}_h)_K^- = 0, \quad (\hat{p}_h)_0^- = 0 \quad (3.16)$$

$$(\hat{p}_h)_0^- = (\hat{p}_h)_0^+, \quad (\hat{p}_h)_K^- = (\hat{p}_h)_K^+ \quad (3.17)$$

are used. Then the LDG formulation (2.9) satisfies the L^2 -stability with the numerical fluxes defined in (2.10)-(2.11). \square

4. Numerical illustrations

To illustrate the efficiency of the LDG method and θ -family of the approximation over the numerical solutions of the Burgers equation, let us take into consideration the following two challenging problems. The problems have homogenous Dirichlet boundary conditions with $\epsilon \ll 1$. The produced results have been compared with the literature [14, 4, 17, 27, 25, 20, 1, 10, 18, 19] and exact solution. To evaluate error norms of the present results we prefer to use the following norm definitions,

$$\|E\|_{L_\infty} = \max_i |u_i^{\text{numerical}} - u_i^{\text{exact}}|$$

$$\|E\|_{L_2} = \sqrt{h \sum_{i=0}^K (u_i^{\text{numerical}} - u_i^{\text{exact}})^2}.$$

Throughout numerical calculations, pointwise errors have been computed with the use of following definition

$$E(x_i) = |u_i^{\text{numerical}} - u_i^{\text{exact}}|.$$

Example 1 [25, 29]. Consider the Burgers equation (1.1) with initial condition

$$u(x, 0) = \sin(\pi x), \quad 0 \leq x \leq 1 \quad (4.1)$$

and boundary conditions

$$u(0, t) = 0, \quad u(1, t) = 0, \quad t > 0. \quad (4.2)$$

Exact solution of (1.1) under the consideration of cases (4.1)-(4.2) given by Cole [9] is

$$u(x, t) = 2\pi\epsilon \frac{\sum_{n=1}^{\infty} a_n \exp(-n^2\pi^2\epsilon t) n \sin(n\pi x)}{a_0 + \sum_{n=1}^{\infty} a_n \exp(-n^2\pi^2\epsilon t) \cos(n\pi x)} \quad (4.3)$$

with the Fourier coefficients

$$a_0 = \int_0^1 \exp\left\{-(2\pi\epsilon)^{-1} [1 - \cos(\pi x)]\right\} dx, \quad (4.4)$$

$$a_n = 2 \int_0^1 \exp\left\{-(2\pi\epsilon)^{-1} [1 - \cos(\pi x)]\right\} \cos(n\pi x) dx. \quad (4.5)$$

Comparison of the computed results with solutions of a finite difference technique [20] has been carried out for parameters $t = 0.5$ and $\epsilon = 1$ as in Table 1. For the corresponding results, various values of the flux parameter are considered as $\alpha = 0$, $\alpha = 0.5$ and $\alpha = 1$. Use of the central flux, $\alpha = 0.5$, has been seen to produce more accurate results among the present cases. All the present cases have been observed to produce more accurate results than the literature [20] even if fewer elements in space is accepted.

By considering L_∞ and L_2 errors, effects of the polynomial degree of N have been presented in Table 2 for $\epsilon = 0.1$, $dt = 0.001$, $K = 20$, $t = 2.3$ and $\alpha = 0.5$.

As realized from the table, the accuracy grows uniformly when the polynomial order increases and even when we consider $N = 2$ and fewer spatial elements, the LDG solutions have been seen to be more accurate than the literature [20].

In Figures 1-3, the absolute error norms, $\epsilon_i = |u_i^{numerical} - u_i^{exact}|$, of the considered numerical approaches are compared with changing values of the flux parameter α , polynomial degree of N and viscosity constant ϵ , respectively. As seen in Figures 1-3, the present method has ability to capture high accuracy even if various problem parameters are accepted.

The results presented in Table 3 have been compared with the results of a sixth order compact finite difference scheme [25]. The upwind flux parameter is suggested to be $\alpha = 0$, when the advection is dominant, $\epsilon = 0.01$ (see Table 3). The parameter values are taken to be $dt = 0.01$ and $K = 20$ in the table for various time values up to $t = 4$. In the comparison, even fewer elements are used both in space and time, the present algorithms have been observed to produce more accurate results than the literature [25].

Yet, a careful discussion has been done for advection dominant cases of the physical mechanism, $\epsilon = 0.004$ and $\epsilon = 0.003$ (see Tables 4-5). Two different flux parameter values, $\alpha = 0$ and $\alpha = 1$, are used with cubic and quartic interpolation polynomials. The currently produced solutions are compared with the wavelet-based solution given in [14]. Even if fewer elements are considered in both time and space, more accurate results than the literature [14] are usually observed to be found.

The LDG solutions have been compared with solutions of various versions of finite element techniques [1, 10] (see Table 6). In the table, we consider the central flux, $\alpha = 0.5$, and quadratic, cubic and quartic interpolation polynomials to compute accurate solutions of the problem through L_∞ and L_2 errors. As seen in the table, the present results with the use of less number of elements are more accurate than the other results.

Note that responses of the mechanism governed by the Burgers equation for various values of the viscosity constant ϵ were discussed in the literature [23, 24, 25]. Physical behaviour of the natural process analysed by the LDG in terms of the viscosity parameter exhibits the expected characteristics of our problem. By taking into account the parameters as $\epsilon = 0.001$, $h = 0.0016$, $dt = 0.1$, $\alpha = 0.5$ and $N = 5$, the steep behaviour of the considered problem has been presented as seen in Figure 4.

TABLE 1. Comparison of the results for various α values with $t = 0.5$, $\varepsilon = 1$ and $dt = 0.001$

x	$N = 3$			[20]	Exact
	$K = 20$			BDF3	
	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$	$K = 100$	
0.1	0.0022131	0.0022130	0.0022133	0.002213	0.0022130
0.2	0.0042103	0.0042100	0.0042105	0.004209	0.0042101
0.3	0.0057964	0.0057960	0.0057966	0.005795	0.0057961
0.4	0.0068162	0.0068158	0.0068164	0.006815	0.0068159
0.5	0.0071696	0.0071691	0.0071696	0.007168	0.0071692
0.6	0.0068211	0.0068206	0.0068211	0.006820	0.0068207
0.7	0.0058043	0.0058038	0.0058042	0.005803	0.0058039
0.8	0.0042182	0.0042178	0.0042181	0.004217	0.0042178
0.9	0.0022181	0.0022178	0.0022179	0.002218	0.0022178
L_∞	2.78E-07	2.92E-07	2.71E-07		
L_2	2.15E-07	5.15E-08	2.33E-07		

TABLE 2. Comparison of the results for various N values with $t = 2.3$, $\varepsilon = 0.1$ and $dt = 0.001$

x	$\alpha = 0.5$				$K = 100$	Exact
	$K = 20$				BDF3	
	$N = 2$	$N = 3$	$N = 4$	$N = 5$	[20]	
0.1	0.0221386	0.0221400	0.0221396	0.0221396	0.02253	0.0221396
0.2	0.0427946	0.0427961	0.0427956	0.0427956	0.04357	0.0427956
0.3	0.0604302	0.0604318	0.0604312	0.0604312	0.06155	0.0604313
0.4	0.0734421	0.0734438	0.0734431	0.0734431	0.07485	0.0734431
0.5	0.0802301	0.0802317	0.0802309	0.0802309	0.08182	0.0802310
0.6	0.0793984	0.0793996	0.0793988	0.0793988	0.08104	0.0793988
0.7	0.0701070	0.0701075	0.0701068	0.0701068	0.07161	0.0701068
0.8	0.0525208	0.0525205	0.0525198	0.0525198	0.05368	0.0525198
0.9	0.0281758	0.0281746	0.0281740	0.0281740	0.02881	0.0281740
L_∞	1.78E-06	7.70E-07	1.72E-08	1.50E-08		
L_2	6.82E-07	4.33E-07	8.72E-09	7.56E-09		

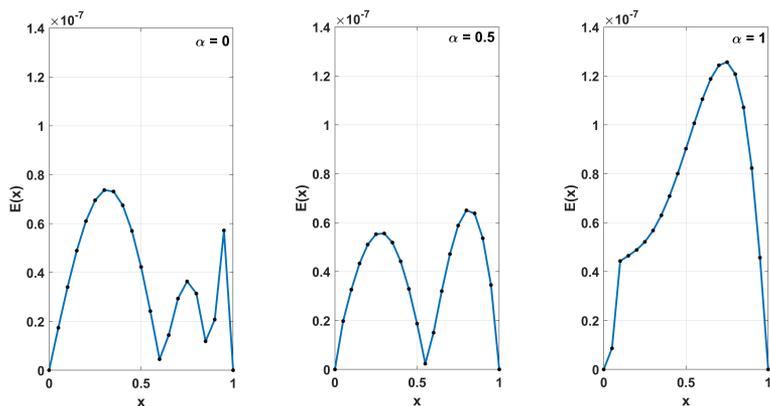


FIGURE 1. Absolute errors for various α values with $h = 0.05$, $dt = 0.0005$, $N = 4$, $t = 0.5$ and $\epsilon = 0.1$.

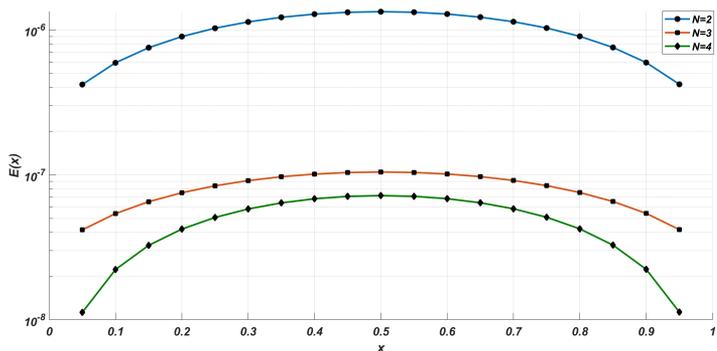


FIGURE 2. Absolute errors for various N values with $h = 0.05$, $dt = 0.0005$, $\alpha = 0.5$, $t = 0.5$ and $\epsilon = 1$.

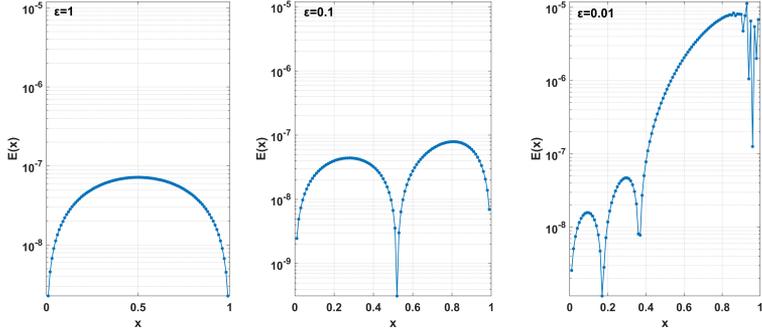


FIGURE 3. Absolute errors for various ϵ values with $h = 0.01$, $dt = 0.0005$, $\alpha = 0.5$, $t = 0.5$ and $N = 5$.

TABLE 3. Comparison of the results for various N values with $\epsilon = 0.01$ and $dt = 0.01$

x	t	$\alpha = 0$			$dt = 0.001$	Exact
		$K = 20$			$K = 100$	
		$N = 3$	$N = 4$	$N = 5$	[25]	
0.10	0.50	0.12113619	0.12114075	0.12114342	0.12114	0.12114353
	2.00	0.04296375	0.04296356	0.04296377	0.04295	0.04296378
	4.00	0.02310437	0.02310418	0.02310423	0.02310	0.02310423
0.30	0.50	0.36033439	0.36027550	0.36027130	0.36027	0.36027106
	2.00	0.12884456	0.12883946	0.12883987	0.12882	0.12883989
	4.00	0.06930920	0.06930816	0.06930828	0.06930	0.06930829
0.50	0.50	0.58771250	0.58864659	0.58869195	0.58870	0.58869577
	2.00	0.21456748	0.21455796	0.21455806	0.21455	0.21455805
	4.00	0.11549709	0.11549459	0.11549475	0.11549	0.11549476
0.70	0.50	0.78821499	0.79299892	0.79344182	0.79354	0.79349341
	2.00	0.29997904	0.29999746	0.29999771	0.29999	0.29999777
	4.00	0.16121542	0.16121454	0.16121465	0.16121	0.16121465
0.90	0.50	0.92582505	0.93663105	0.93793355	0.93822	0.93810831
	2.00	0.37286254	0.37327919	0.37327588	0.37328	0.37327763
	4.00	0.16595137	0.16605901	0.16605864	0.16605	0.16605872
L_∞		9.83E-04	4.92E-05	3.83E-06		
L_2		2.20E-04	1.11E-05	8.58E-07		

TABLE 4. Comparison of the results for various values of the parameters N and α with $\varepsilon = 0.004$ and $dt = 0.01$.

x	t	$K = 40$				$K = 100$	Exact
		$N = 3$		$N = 4$		[14]	
		$\alpha = 0$	$\alpha = 1$	$\alpha = 0$	$\alpha = 1$	$dt = 0.001$	
0.25	5	0.046972404	0.04697755	0.04697220	0.04697232	0.04697	0.04697225
	10	0.024219379	0.02422110	0.02421933	0.02421933	0.02422	0.02421935
	15	0.016315418	0.01631626	0.01631540	0.01631541	0.01632	0.01631540
0.50	5	0.093938473	0.09394420	0.09393796	0.09393811	0.09394	0.09393781
	10	0.048437298	0.04843974	0.04843717	0.04843723	0.04843	0.04843716
	15	0.032594629	0.03259594	0.03259459	0.03259462	0.03259	0.03259459
0.75	5	0.140883529	0.14088798	0.14088513	0.14088532	0.14089	0.14088686
	10	0.072202345	0.07220549	0.07220248	0.07220255	0.07221	0.07220247
	15	0.046775023	0.04677717	0.04677530	0.04677534	0.04678	0.04677529
L_∞		3.33E-06	6.39E-06	1.72E-06	1.53E-06		
L_2		5.40E-07	1.54E-06	2.74E-07	2.48E-07		

TABLE 5. Comparison of the results for various values of the parameters N and α with $\varepsilon = 0.003$ and $dt = 0.01$.

x	t	$K = 40$				$K = 100$	Exact
		$N = 3$		$N = 4$		[14]	
		$\alpha = 0$	$\alpha = 1$	$\alpha = 0$	$\alpha = 1$	$dt = 0.001$	
0.25	5	0.04698137	0.04698827	0.04698091	0.04698113	0.04697	0.04697225
	10	0.02422182	0.02422423	0.02422172	0.02422178	0.02422	0.02421935
	15	0.01631716	0.01631835	0.01631711	0.01631714	0.01632	0.01631540
0.50	5	0.09395587	0.09396301	0.09395531	0.09395556	0.09394	0.09393781
	10	0.04844325	0.04844657	0.04844300	0.04844310	0.04843	0.04843716
	15	0.03263180	0.03263361	0.03263170	0.03263175	0.03259	0.03259459
0.75	5	0.14090425	0.14090646	0.14091177	0.14091239	0.14089	0.14088686
	10	0.07260301	0.07260674	0.07260292	0.07260305	0.07221	0.07220247
	15	0.04838631	0.04838885	0.04838641	0.04838648	0.04678	0.04677529
L_∞		1.21E-05	9.87E-06	4.57E-06	3.94E-06		
L_2		1.92E-06	2.52E-06	7.22E-07	6.26E-07		

TABLE 6. Comparison of the results for various N values with $\varepsilon = 0.01$ and $dt = 0.001$.

x	t	$N = 2$	$N = 3$	$N = 4$			Exact
		$\alpha = 0.5$					
		$K = 20$			[20]	[1]	
0.25	0.4	0.30891550	0.30889775	0.30889399	0.30890	0.30891	0.30889423
	0.6	0.24075573	0.24074158	0.24073888	0.24074	0.24075	0.24073902
	0.8	0.19568868	0.19567740	0.19567548	0.19568	0.19568	0.19567557
	1.0	0.16257556	0.16256622	0.16256480	0.16257	0.16257	0.16256486
	3.0	0.02720591	0.02720270	0.02720231	0.02720	0.02721	0.02720231
0.50	0.4	0.56964355	0.56963414	0.56963246	0.56964	0.56969	0.56963245
	0.6	0.44721888	0.44720693	0.44720547	0.44721	0.44723	0.44720552
	0.8	0.35924901	0.35923722	0.35923601	0.35924	0.35926	0.35923606
	1.0	0.29192816	0.29191701	0.29191591	0.04020	0.04021	0.04020492
0.75	0.4	0.62544987	0.62543312	0.62543823	0.62541	0.62543	0.62543790
	0.6	0.48722718	0.48721269	0.48721519	0.48719	0.48723	0.48721497
	0.8	0.37393607	0.37392129	0.37392186	0.37390	0.37394	0.37392175
	1.0	0.28748959	0.28747493	0.28747444	0.28746	0.28750	0.28747441
	3.0	0.02977639	0.02977258	0.02977212	0.02977	0.02978	0.02977213
L_∞		1.63E-05	2.61E-06	2.17E-07			
L_2		9.74E-06	1.43E-06	7.65E-08			

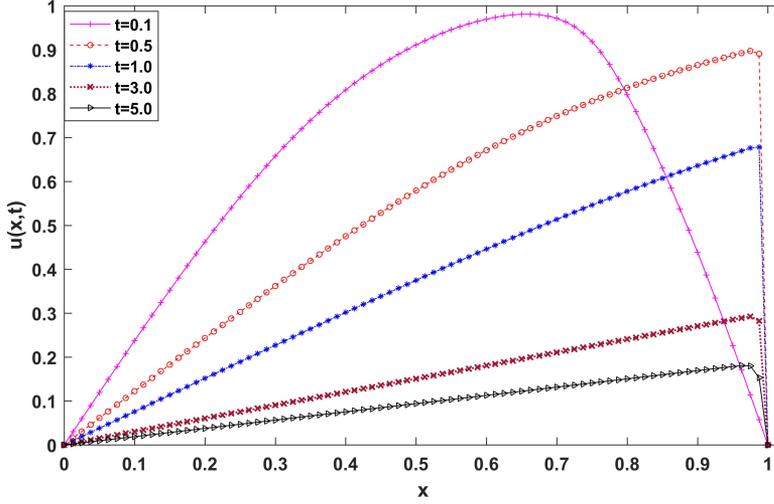


FIGURE 4. Numerical solutions of Example 1 at different times for $\epsilon = 0.001$, $h = 0.0016$, $dt = 0.1$, $\alpha = 0.50$ and $N = 5$.

Example 2 [25, 29]. Let us now take the Burgers equation (1.1) with the initial condition

$$u(x, 0) = 4x(1 - x), \quad 0 \leq x \leq 1 \quad (4.6)$$

and the boundary conditions

$$u(0, t) = 0, \quad u(1, t) = 0, \quad t > 0. \quad (4.7)$$

Exact solution of equation (1.1) under the consideration of cases (4.6)-(4.7) given by [9] has the form (4.3) with the Fourier coefficients

$$a_0 = \int_0^1 \exp \left\{ -x^2 (3\epsilon)^{-1} (3 - 2x) \right\} dx, \quad (4.8)$$

$$a_n = 2 \int_0^1 \exp \left\{ -x^2 (3\epsilon)^{-1} (3 - 2x) \right\} \cos(n\pi x) dx. \quad (4.9)$$

Comparison of the currently computed solutions with the exact solution have been done for the kinematic viscosity, $\epsilon = 1$. The computed results in Table 7 are seen to be more accurate than the literature [27, 18]. The derived LDG has been seen to be more effective than a different version of the discontinuous Galerkin method of Shao et. al. [27]. Note that their approach are based on a formulation involving the Legendre polynomials and the Runge-Kutta method with the only case of $\alpha = 0$. To produce the results of the LDG, even less number of elements, comparison to the corresponding references, have been used in time. The results in Table 7 have been presented for various degrees of polynomial with the central flux parameter $\alpha = 0.5$.

Absolute errors of the current approaches are compared with changing values of the flux parameter α , polynomial degree of N and viscosity constant ϵ , respectively, as seen in Figures 5-7. As realized from the figures, the LDG is capable of capturing highly accurate response even for advection-dominant case.

Comparison of the currently produced results with the literature [17, 18] and exact solution has been given in Table 8. The current solutions are more accurate than the corresponding ones in literature when the advection is relatively more dominant to the diffusion, $\epsilon = 0.01$, and $h = 0.0125$. By considering quadratic, cubic and quartic interpolation polynomials with the central flux parameter, $\alpha = 0.5$, the produced results are seen to require less effort in time comparison to the corresponding references. For the choice of the parameters $\epsilon = 0.01$, $dt = 0.002$ and $h = 0.01$ and higher degree polynomials with upwind flux selection $\alpha = 0$; the computed results have been exhibited in a comparative way with finite difference [25] and boundary element [4] methods as well as the exact solution in Table 9. The produced solutions revealed that less time effort and also less computational time are needed to catch high accuracy in comparison with the previously mentioned effective methods.

To show the efficiency of the current method, we compare the results against the ones obtained by an analytical solution and by the method presented in [6]. We consider $dt = 0.01$, $h = 0.025$ for $\epsilon = 0.003$ and $\epsilon = 0.004$ in Tables 10-11, respectively, with upwind flux selections, i.e. $\alpha = 0$ and $\alpha = 1$. As underlined for various times, even less number of time elements are seen to be enough to have highly accurate solutions.

TABLE 7. Comparison of the results for various N values with $\epsilon = 1$ and $h = 0.0002$.

x	t	$N = 2$	$N = 3$	$N = 4$	[18]	[27]	Exact
		$\alpha = 0.5$			$dt = 0.0001$		
			$K = 20$		$h = 0.0125$	$N = 4$	
0.25	0.05	0.42630195	0.42628824	0.42628541	0.42629	0.4262864	0.42628562
	0.10	0.26149427	0.26148187	0.26147970	0.26149	0.2614801	0.26147981
	0.15	0.16148963	0.16147922	0.16147753	0.16148	0.1614777	0.16147762
	0.25	0.06109465	0.06108847	0.06108753	0.06109	0.0610875	0.06108758
0.50	0.05	0.62809374	0.62808576	0.62808368	0.62809	0.6280846	0.62808373
	0.10	0.38343689	0.38342459	0.38342229	0.38343	0.3834228	0.38342242
	0.15	0.23406902	0.23405720	0.23405522	0.23406	0.2340554	0.23405533
	0.25	0.08724142	0.08723381	0.08723263	0.08724	0.0872327	0.08723270
0.75	0.05	0.46526695	0.46525478	0.46525263	0.46526	0.4652528	0.46525262
	0.10	0.28158776	0.28157474	0.28157258	0.28158	0.2815727	0.28157264
	0.15	0.16975116	0.16973999	0.16973820	0.16974	0.1697383	0.16973828
	0.25	0.06229717	0.06229077	0.06228980	0.06229	0.0622898	0.06228985
L_∞		1.63E-05	2.61E-06	2.17E-07			
L_2		9.74E-06	1.43E-06	7.65E-08			

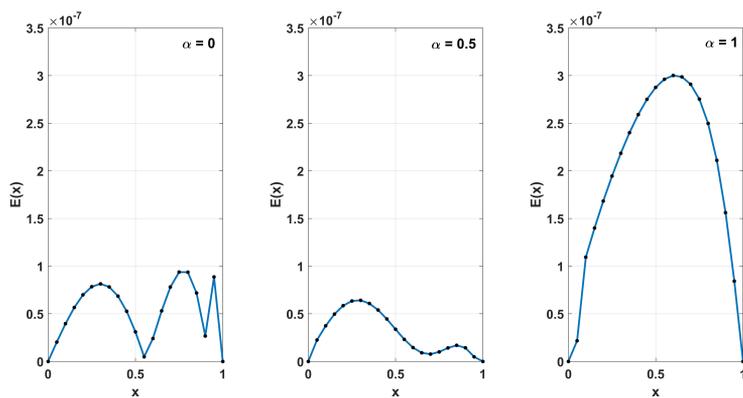


FIGURE 5. Absolute errors for various values of α with $h = 0.05$, $dt = 0.0005$, $N = 4$ and $\epsilon = 0.1$

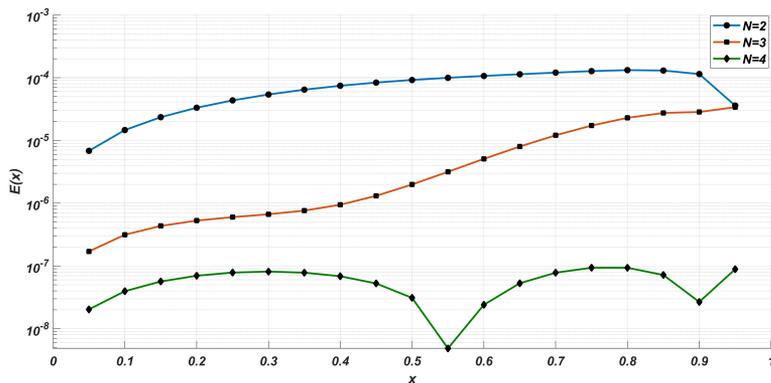


FIGURE 6. Absolute errors for various N values $h = 0.05$, $dt = 0.0005$, $\alpha = 0.5$ and $\epsilon = 0.1$.

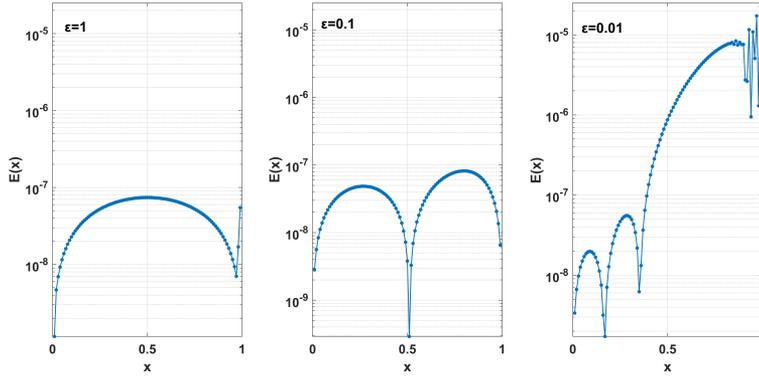


FIGURE 7. Absolute errors for various ϵ values with $h = 0.01$, $dt = 0.0005$, $\alpha = 0.5$ and $N = 5$.

TABLE 8. Comparison of the results for various N values with $\epsilon = 0.01$ and $h = 0.025$.

x	t	$N = 3$	$N = 4$	$N = 5$	$dt = 0.001$		Exact
		$\alpha = 0.5$			[19]	[17]	
			$K = 20$				
0.25	0.4	0.31752739	0.31752261	0.31752262	0.32091	0.31749	0.31752288
	0.6	0.24614156	0.24613830	0.24613831	0.24910	0.24612	0.24613846
	0.8	0.19955748	0.19955521	0.19955522	0.20211	0.19954	0.19955531
	1.0	0.16560022	0.16559857	0.16559858	0.16782	0.16559	0.16559863
	3.0	0.02775913	0.02775871	0.02775871	0.02828	0.02776	0.02775871
0.50	0.4	0.58453985	0.58453728	0.58453731	0.58788	0.58448	0.58453726
	0.6	0.45797847	0.45797634	0.45797636	0.46174	0.45793	0.45797640
	0.8	0.36739983	0.36739812	0.36739815	0.37111	0.36736	0.36739819
	1.0	0.29834451	0.29834305	0.29834307	0.30183	0.29831	0.29834311
	3.0	0.04106555	0.04106498	0.04106498	0.04185	0.04106	0.04106499
0.75	0.4	0.64561023	0.64561574	0.64561586	0.65054	0.64547	0.64561551
	0.6	0.50267355	0.50267593	0.50267599	0.50825	0.50255	0.50267575
	0.8	0.38533533	0.38533560	0.38533564	0.39068	0.38523	0.38533552
	1.0	0.29585750	0.29585671	0.29585673	0.30057	0.29578	0.29585668
	3.0	0.03044014	0.03043964	0.03043964	0.03106	0.03044	0.03043965
L_∞		5.27E-06	2.75E-07	3.49E-07			
L_2		2.08E-06	1.05E-07	1.23E-07			

TABLE 9. Comparison of the results for various N values with $\varepsilon = 0.01$ and $h = 0.025$.

x	t	$N = 4$	$N = 5$	$N = 6$	$dt = 0.001$		Exact
		$\alpha = 0$			[4]	[25]	
		$K = 40$					
0.10	0.50	0.12846195	0.12846208	0.12846208	0.12808	0.12846	0.12846216
	2.00	0.04381384	0.04381385	0.04381385	0.04388	0.04379	0.04381385
	4.00	0.02334500	0.02334500	0.02334500	0.02351	0.02334	0.02334500
0.30	0.50	0.37848952	0.37848934	0.37848934	0.37956	0.37849	0.37848913
	2.00	0.13134517	0.13134519	0.13134519	0.13129	0.13131	0.13134519
	4.00	0.07002717	0.07002718	0.07002718	0.07009	0.07002	0.07002718
0.50	0.50	0.60988033	0.60988260	0.60988263	0.61768	0.60991	0.60988613
	2.00	0.21858803	0.21858804	0.21858804	0.21873	0.21858	0.21858801
	4.00	0.11668202	0.11668202	0.11668202	0.11671	0.11667	0.11668202
0.70	0.50	0.80973794	0.80976141	0.80976228	0.83022	0.80986	0.80978166
	2.00	0.30534807	0.30534808	0.30534808	0.30614	0.30534	0.30534815
	4.00	0.16287830	0.16287830	0.16287830	0.16293	0.16287	0.16287830
0.90	0.50	0.94590664	0.94597833	0.94598192	0.98068	0.94615	0.94601416
	2.00	0.38027297	0.38027291	0.38027294	0.38163	0.38027	0.38027365
	4.00	0.16857740	0.16857740	0.16857740	0.16766	0.16857	0.16857741
L_∞		1.08E-04	3.65E-05	3.29E-05			
L_2		1.85E-05	6.62E-06	6.06E-06			

TABLE 10. Comparison of the results for various values of the polynomial degree N and α with $\varepsilon = 0.003$ and $dt = 0.01$.

x	t	$K = 40$				$K = 100$	Exact
		$N = 3$		$N = 4$		[14]	
		$\alpha = 0$	$\alpha = 1$	$\alpha = 0$	$\alpha = 1$	$dt = 0.001$	
0.25	5	0.04746536	0.04747324	0.04746470	0.04746502	0.04746	0.04746474
	10	0.02434983	0.02435249	0.02434968	0.02434976	0.02434	0.02434970
	15	0.01637512	0.01637643	0.01637506	0.01637509	0.01637	0.01637507
0.50	5	0.09491255	0.09492120	0.09491167	0.09491209	0.09491	0.09491170
	10	0.04869850	0.04870229	0.04869814	0.04869830	0.04870	0.04869814
	15	0.03274766	0.03274969	0.03274751	0.03274759	0.03274	0.03274752
0.75	5	0.14231142	0.14231499	0.14231903	0.14231987	0.14232	0.14232395
	10	0.07298613	0.07299050	0.07298591	0.07298610	0.07298	0.07298597
	15	0.04856832	0.04857117	0.04856835	0.04856845	0.04857	0.04856845
L_∞		1.25E-05	9.51E-06	4.91E-06	4.08E-06		
L_2		1.99E-06	2.75E-06	7.77E-07	6.51E-07		

TABLE 11. Comparison of the results for various values of the polynomial degree N and α with $\varepsilon = 0.004$ and $dt = 0.01$.

x	t	$K = 40$				$K = 100$	Exact
		$N = 3$		$N = 4$		[14]	
		$\alpha = 0$	$\alpha = 1$	$\alpha = 0$	$\alpha = 1$	$dt = 0.001$	
0.25	5	0.04743883	0.04744470	0.04743852	0.04743869	0.04744	0.04743858
	10	0.02434269	0.02434459	0.02434261	0.02434266	0.02434	0.02434263
	15	0.01637127	0.01637219	0.01637124	0.01637126	0.01637	0.01637125
0.50	5	0.09486174	0.09486863	0.09486104	0.09486128	0.09491	0.09486089
	10	0.04868331	0.04868610	0.04868313	0.04868321	0.04868	0.04868313
	15	0.03270706	0.03270853	0.03270699	0.03270703	0.03270	0.03270700
0.75	5	0.14224513	0.14225088	0.14224663	0.14224693	0.14224	0.14224850
	10	0.07258099	0.07258462	0.07258105	0.07258116	0.07258	0.07258104
	15	0.04696414	0.04696653	0.04696437	0.04696443	0.04696	0.04696437
L_∞		3.37E-06	7.74E-06	1.87E-06	1.57E-06		
L_2		5.53E-07	1.85E-06	2.97E-07	2.57E-07		

5. Concluding remarks and recommendations

In this paper, a hybrid numerical approach based on a local discontinuous Galerkin (LDG) method for spatial discretization and θ -family for time integration has been presented for dealing with the behavior of nonlinear advection-diffusion processes. Use of the Gauss-Lobatto grid points has been seen to be very effective for p -refinement of the solution. It has been proved that the currently derived technique is unconditionally stable. The numerical and theoretical results to the parameter based numerical fluxes of the LDG formulation have been successfully produced. Considering two challenging examples of the Burgers equation, efficiency of the current numerical technique over the rivals has been demonstrated for both advection and diffusion dominated physical systems. The central flux selection has been found to be optimum for accurate solutions. For further research, the current approach can also be applied to more realistic but much more strenuous and complex physical environments.

Acknowledgement

The first author would like to thank the Science Fellowships and Grant Programmes Department of TUBITAK (BIDEB) for their support to his academic research.

References

- [1] E. N. Aksan, Quadratic B-spline finite element method for numerical solution of the Burgers equation, *Applied Mathematics and Computation* **174** (2006), 884-896.
- [2] P. Antonietti, M. Verani, C. Vergarab and S. Zonca, Numerical solution of fluid-structure interaction problems by means of a high order Discontinuous Galerkin method on polygonal grids, *Finite Elements in Analysis and Design* **159** (2019), 1-14.
- [3] A. Asaithambi, Numerical solution of the Burgers' equation by automatic differentiation, *Applied Mathematics and Computation* **216** (2010), 2700-2708.
- [4] A. R. Bahadir and M. Saglam, A mixed finite difference and boundary element approach to one-dimensional Burgers' equation, *Applied Mathematics and Computation* **160** (2005), 663-673.
- [5] H. Bateman, Some recent researches on the motion of fluids, *Monthly Weather Review* **43** (1915), 163-170.
- [6] J. M. Burgers, A mathematical model illustrating the theory of turbulence, *Advances in Applied Mathematics* **1** (1948), 171-199.
- [7] B. Cockburn and C. W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, *SIAM Journal on Numerical Analysis* **35** (1998), 2440-2463.
- [8] B. Cockburn and C. W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: General framework, *Mathematics of Computation* **52** (1989), 411-435.
- [9] J. D. Cole, On a quasi-linear parabolic equation in aerodynamics, *Quarterly of Applied Mathematics* **9** (1951), 225-236.
- [10] I. Dag, B. Saka and A. Boz, B-spline Galerkin methods for numerical solutions of Burgers' equation, *Applied Mathematics and Computation* **166** (2005), 506-522.
- [11] U. Erdogan, M. Sari and H. Kocak, Efficient numerical treatment of nonlinearities in the advection-diffusion-reaction equations, *International Journal of Numerical Methods for Heat & Fluid Flow* **29** (2019), No.1, 132-145.
- [12] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms. Analysis and Applications*, New York, USA: Springer, 2008.
- [13] E. Hopf, The partial differential equation $u_t + uu_x = \epsilon u_{xx}$, *Communications on Pure and Applied Mathematics* **9** (1950), 201-230.
- [14] R. Jiwari, A hybrid numerical scheme for the numerical solution of the Burgers' equation, *Computer Physics Communications* **188** (2015), 50-67.
- [15] R. Jiwari, A Haar wavelet quasilinerization approach for numerical simulation of Burgers' equation, *Computer Physics Communications* **183** (2012), 2413-2423.
- [16] A. Korkmaz, Shock wave simulations using sinc differential quadrature method, *International Journal of Computer Aided Engineering and Technology* **26** (2011), No.6, 654-674.
- [17] S. Kutluay and A. Esen, A lumped Galerkin method for solving the Burgers equation, *International Journal of Computer Mathematics* **81** (2004), No.11, 1433-1444.
- [18] S. Kutluay, A. R. Bahadir and A. Ozdes, Numerical solution of one-dimensional Burgers' equation: explicit and exact-explicit finite difference methods, *Journal of Computational and Applied Mathematics* **103** (1999), 251-261.
- [19] S. Kutluay, A. Esen and I. Dag, Numerical solutions of the Burgers' equation by the least-squares quadratic B-spline finite element method, *Journal of Computational and Applied Mathematics* **167** (2004), 21-33.
- [20] V. Mukundan and A. Awasthi, Efficient numerical techniques for Burgers' equation, *Applied Mathematics and Computation* **262** (2015), 282-297.

- [21] W. Reed and T. Hill, Triangular mesh methods for the neutron transport equation, Los Alamos Report LA-UR-73-479, 1973.
- [22] Z. Rong-Pei, Y. Xi-Yun and Z. Guo-Zhong, Local discontinuous Galerkin method for solving Burgers and coupled Burgers equations, *Chinese Physics B* **20** (2011), No.11, 110205-110210.
- [23] M. Sari and H. Tunc, An optimization technique in analysing the Burgers equation, *Sigma Journal of Engineering and Natural Sciences* **35** (2017), No.2, 369-386.
- [24] M. Sari, H. Tunc and M. Seydaoglu, Higher order splitting approaches in analysis of the Burgers equation, *Kuwait Journal of Science* **46** 2019, No.1, 1-14.
- [25] M. Sari and G. Gurarslan, A sixth-order compact finite difference scheme to the numerical solutions of Burgers' equation, *Applied Mathematics and Computation* **208** (2009), 475-483.
- [26] M. Seydaoglu, U. Erdogan and T. Ozis, Numerical solution of Burgers' equation with higher order splitting methods, *Journal of Computational and Applied Mathematics* **191** (2016), 410-421.
- [27] L. Shao, X. Feng and Y. He, The local discontinuous Galerkin finite element method for Burgers equation, *Mathematical and Computer Modelling* **54** (2011), 2943-2954.
- [28] I. Touloupoulos, An interior penalty discontinuous Galerkin finite element method for quasilinear parabolic problems, *Finite Elements in Analysis and Design* **95** (2015), 42-50.
- [29] H. Tunc, Various finite element techniques for advection-diffusion-reaction processes, M.Sc Thesis, Yildiz Technical University, Istanbul, Turkey, 2017.

Huseyin Tunc

Department of Mathematics, Yildiz Technical University, Istanbul, Turkey.

E-mail address: f2517005@std.yildiz.edu.tr

Murat Sari

Department of Mathematics, Yildiz Technical University, Istanbul, Turkey.

E-mail address: sarim@yildiz.edu.tr

Received: May 25, 2020; Revised: November 22, 2020; Accepted: December 7, 2020