Mathematical Sciences (2021) 15:21–28
https://doi.org/10.1007/s40096-020-00356-3

ORIGINAL RESEARCH

Numerical treatment for Burgers–Fisher and generalized Burgers–Fisher equations

S. Kumar[1] · S. Saha Ray[1]

Received: 28 June 2020 / Accepted: 20 October 2020 / Published online: 18 November 2020

Abstract

In this paper, the discontinuous Legendre wavelet Galerkin method is proposed for the numerical solution of the Burgers–Fisher and generalized Burgers–Fisher equations. This method combines both the discontinuous Galerkin and the Legendre wavelet Galerkin methods. Various properties of Legendre wavelets have been used to find the variational form of the governing equation. This variational form transforms it into a system of ordinary differential equations which will be solved numerically. Some illustrative examples are presented to emphasize the efficiency and reliability of the proposed method.

Keywords Burgers–Fisher equation · Legendre wavelets · Operational matrix for derivative · Variational Form

Introduction

The study of nonlinear partial differential equation (NPDEs) has attracted many researchers due to its pertinent features in many fields of applied science and engineering. The generalized Burgers–Fisher equation (gBFE) appears in a variety of applications in the fields of fluid dynamics, financial mathematics, heat conduction, turbulence, gas dynamics, and many other fields of applied science and engineering. In this article, the discontinuous Legendre wavelet Galerkin method (DLWGM) has been exerted for solving gBFE numerically. This method is fundamentally based on utilization of Legendre wavelets as wavelet basis [1–4].

The gBFE is defined as [5]

$$u_t + \alpha u^{\delta} u_x - u_{xx} = \beta u(1 - u^{\delta}), \qquad x \in (0,1), \qquad t \geq 0, \tag{1}$$

with the initial condition

$$u(x,0) = \left(\frac{1}{2} + \frac{1}{2}\tanh\left(\frac{-\alpha\delta}{2(\delta+1)}x\right)\right)^{\frac{1}{\delta}}. \tag{2}$$

The exact solution of (1) is given by [5]

$$u_{\text{exact}}(x,t) = \left(\frac{1}{2} + \frac{1}{2}\tanh\left(\frac{-\alpha\delta}{2(\delta+1)}\left(x - \left(\frac{\alpha}{(\delta+1)} + \frac{\beta(\delta+1)}{\alpha}\right)t\right)\right)\right)^{\frac{1}{\delta}}. \tag{3}$$

As stated earlier, the discontinuous Legendre wavelet Galerkin (DLWG) scheme combines the discontinuous Galerkin (DG) and Legendre wavelet Galerkin (LWG) method to systematically find numerical solutions of NPDEs. The DG method is a finite element method which forms the weak formulation of the given problem in the piecewise continuous space of functions. However, the Legendre wavelets (LWs) are element-wise discontinuous at the boundary points of the interval. As a result, this will efficiently enable in the construction of weak formulation for the solution of Eq. (1). We construct the variational form of Eq. (1) by following the technique of DG method where numerical fluxes are used to

✉ S. Saha Ray
santanusaharay@yahoo.com

[1] Department of Mathematics, National Institute Technology Rourkela, Rourkela, Odisha 769008, India

balance with complicated geometries. We will further construct variational form for the initial condition (2). The various properties of LWs are used to transform the variational form of Eq. (1) along with variational form of initial condition (2) into the system of ordinary differential equations (ODEs) that can be solved numerically [3]. In this paper, DLWG method has been successfully used first time ever to obtain the solution of modified Burgers–Fisher equation.

Brief description on LWs

The Rodrigue's form of Legendre polynomial of degree k is given by

$$P_k(x) = \frac{1}{2^k k!}\frac{\mathrm{d}^k}{\mathrm{d}x^k}[(x^2-1)^k]. \tag{4}$$

From Eq. (4), we have

$$\begin{aligned} &P_0(x) = 1,\ \ P_1(x) = x, \\ &P_k(x) = \frac{2k-1}{k}\, x\, P_{k-1}(x) - \frac{k-1}{k}\, P_{k-2}(x), \quad k \geq 2. \end{aligned} \tag{5}$$

The Legendre scale function $\phi_k(x)$ is defined by [6–8]

$$\phi_k(x) \ = \ \begin{cases} \sqrt{2k+1}\, P_k(2x-1), & x \in [0,1) \\ 0, & x \notin [0,1) \end{cases}. \tag{6}$$

For $n = 0, 1, 2, ...,$ and $l = 0, 1, 2, ..., 2^n - 1$, we define the interval I_{nl} by

$$I_{nl} = \left[\frac{l}{2^n}, \frac{(l+1)}{2^n}\right). \tag{7}$$

Now for $p = 1, 2, ...,$ we define a subspace of piecewise polynomial functions as follows:

$V_{p,n} = \{f\ :\ f|_{I_{nl}}$ is a polynomial of degree strictly less than p; and f vanishes elsewhere $\}$.

The orthonormal basis for the subspace $V_{p,0}$ is given by the whole set $\{\phi_k\}_{k=0}^{p-1}$ which forms an orthonormal basis for the subspace of $V_{p,n}$. Generally, the subspace $V_{p,n}$ is spanned by $2^n p$ functions which are obtained from $\{\phi_k\}_{k=0}^{p-1}$ by dilations and translations [3], given as

$$\begin{aligned} V_{p,n} \ :&= V_{p,nl} = \mathrm{span}\{\phi_{k,nl}(x) = 2^{n/2}\phi_k(2^n x - l), \\ &0 \leq k \leq p-1, 0 \leq l \leq 2^n - 1\}, \end{aligned} \tag{8}$$

and

$$V_{p,0} \subset V_{p,1} \subset V_{p,2} \subset \cdots V_{p,n} \subset \cdots .$$

The approximation of a function $f \in L_2([0,1])$ in $V_{p,n}$ is expressed by only scale functions as

$$P_{p,n}f(x) = P_n f(x) = \sum_{l=0}^{2^n-1}\sum_{k=0}^{p-1} s_{k,nl}\phi_{k,nl}(x), \tag{9}$$

where P_n is the finest scale projection of the function f and $s_{k,nl}$ are scale coefficients.

Variational form by DLWG method

We construct the weak formulation of Eq. (1) by the DLWG method. The computational domain is [0,1], which is divided into $N = 2^n$ subintervals given by $I_{nl} = [2^{-n}l, 2^{-n}(l+1)]$. Let $x \in I_{nl}$, now for a specific value of l the numerical solution in space $V_{p,n}$ can be approximated by

$$P_n u(x,t) = \sum_{k=0}^{p-1} c_{k,nl}(t)\phi_{k,nl}(x) = C_l^T(t)\Phi_l(x), \tag{10}$$

where $C_l(t) = [c_{0,nl}, c_{1,nl}, ..., c_{p-1,nl}]^T$. The function $C_l(t)$ is evaluated from the initial conditions and the weak solution form of Eq. (1), and $\Phi_l(x) = [\phi_{0,nl}(x), \phi_{1,nl}(x), \ldots, \phi_{p-1,nl}(x)]^T$ is the vector of the LWs basis.

Let u_l^+ and u_l^- be the values of u at $x_l = 2^{-n}l$, $l = 0, 1, 2, \ldots, 2^n - 1$ from right and left, respectively [3]

$$u_l^+ = \lim_{\epsilon \to 0^+} u(x_l + \epsilon), \quad u_l^- = \lim_{\epsilon \to 0^+} u(x_l - \epsilon), \tag{11}$$

where at each boundary point of I_{nl}, $\{u\} = (u^+ + u^-)/2$ and $[[u]] = u^+ - u^-$ represent the mean and jump of function u respectively.

Now for the approximate solution $u_h \in V_{p,n}$, weak formulation of Eqs. (1) and (2) is designed by multiplying Eqs. (1) and (2) by all test functions v_h of subspace $V_{p,n}$ and integrated over each element $I_{n,l}$ and thus we obtain the following

$$\begin{aligned} &\int_{I_{n,l}} \frac{\partial u_h}{\partial t} v_h \mathrm{d}x + \alpha \int_{I_{n,l}} u_h^\delta \frac{\partial u_h}{\partial x} v_h \mathrm{d}x - \int_{I_{n,l}} \frac{\partial^2 u_h}{\partial x^2} v_h \mathrm{d}x \\ &\quad = \int_{I_{n,l}} \beta u_h (1 - u_h^\delta) v_h \mathrm{d}x, \end{aligned} \tag{12}$$

$$\int_{I_{n,l}} u_h(x,0) v_h \mathrm{d}x = \int_{I_{n,l}} u(x,0) v_h \mathrm{d}x, \tag{13}$$

where $l = 0, 1, 2, \ldots, 2^n - 1$. Now simplifying the above equations by formal integration by parts, we obtain

$$\begin{aligned} &\int_{I_{n,l}} \frac{\partial u_h}{\partial t} v_h \mathrm{d}x - \int_{I_{n,l}} \left(\alpha \frac{u_h^{\delta+1}}{\delta+1} - \frac{\partial u_h}{\partial x}\right) \frac{\partial v_h}{\partial x} \mathrm{d}x + (\hat{f})_{x_{l+1}} v_{h(l+1)}^- - (\hat{f})_{x_l} v_{h(l)}^+ \\ &\quad - (\hat{q})_{x_{l+1}} v_{h(l+1)}^- + (\hat{q})_{x_l} v_{h(l)}^+ = \beta \int_{I_{n,l}} u_h (1 - u_h^\delta) v_h \mathrm{d}x, \end{aligned} \tag{14}$$

for each subinterval $I_{n,l}$. The functions $\hat{f}$ and $\hat{q}$ in Eq. (14) are convection and diffusion numerical fluxes, respectively, which are single-valued functions defined at the boundary points of the subinterval. Since the function u_h is discontinuous at the boundary points of the subinterval; therefore, the nonlinear convection and diffusion fluxes are replaced by the numerical fluxes $\hat{f}$ and $\hat{q}$, which arise from integration by parts. Proper estimation of numerical fluxes is essential for the stability of the DLWG method. In this paper, the convection flux $\hat{f}$ is decided to be the local Lax–Friedrichs flux which is given by [3]

$$\hat{f} = \frac{1}{2}\alpha\left(\frac{(u_h^+)^{\delta+1}}{\delta+1} + \frac{(u_h^-)^{\delta+1}}{\delta+1} - (u_h^+ - u_h^-)\right). \tag{15}$$

Also, the diffusion flux $\hat{q}$ [4] is determined as

$$\hat{q} = 2^n\beta_0\mu[[u_h]] + \left\{\frac{\partial u_h}{\partial x}\right\} + 2^{-n}\beta_1\left[\left[\frac{\partial^2 u_h}{\partial x^2}\right]\right]. \tag{16}$$

The selection of numerical fluxes in DLWG method serves the similar idea as those for the LDG method [3].

The variational form computation

In the present analysis, we separately assess each term of Eq. (14) obtained in section "Variational form by DLWG method" by using the characteristics of the LW basis.

The LWs operational matrix of derivative

Let us define the derivative operator $D_{p,n} : V_{p,n} \to V_{p,n}$ for some fixed resolution level n. Let us consider $P_{p,n}u, D_{p,n}u \in V_{p,n}$ with expansions by the Legendre scale functions. Denoting $D_{p,n}\Phi_l(x) = \dot{\Phi}_l(x)$, our aim is to find the $p \times p$ derivative operational matrix R which satisfies [3]

$$\dot{\Phi}_l(x) = R.\Phi_l(x), \tag{17}$$

where $\Phi_l(x) = [\phi_{0,nl}, \phi_{1,nl} ... \phi_{p-1,nl}]$.

The matrix R is calculated as [3]:

$$\begin{aligned}(R)_{i+1,j+1} &= \int_{I_{n,l}} \phi_{i,nl}\frac{\mathrm{d}\phi_{j,nl}}{\mathrm{d}x}\mathrm{d}x \\ &= \int_{\frac{l}{2^n}}^{\frac{l+1}{2^n}} \phi_{i,nl}\frac{\mathrm{d}\phi_{j,nl}}{\mathrm{d}x}\mathrm{d}x \\ &= 2^n\int_0^1 \phi_{i,nl}\frac{\mathrm{d}\phi_{j,nl}}{\mathrm{d}x}\mathrm{d}x \\ &= 2^n(r_0)_{i+1,j+1},\end{aligned} \tag{18}$$

where matrix r_0 is given by

$$(r_0)_{i+1,j+1} = \int_0^1 \phi_{i,nl}\frac{\mathrm{d}\phi_{j,nl}}{\mathrm{d}x}\mathrm{d}x. \tag{19}$$

Now matrix r_0 is calculated by using the Legendre polynomial relation which is given by

$$(2j+1)P_{j(x)} = P'_{j+1}(x) - P'_{j-1}(x). \tag{20}$$

By using above equation we get

$$\begin{aligned}\frac{\phi'_j(x)}{2\sqrt{2j+1}} &= \sqrt{2j-1}\phi_{j-1}(x) + \sqrt{2j-5}\phi_{j-3}(x) + \cdots \\ &\quad + \begin{cases}\phi_0(x), & j \text{ odd} \\ \sqrt{3}\phi_1(x), & j \text{ even}\end{cases}.\end{aligned} \tag{21}$$

Substituting Eq. (21) in Eq. (18) we get r_0 matrix which is represented by

$$(r_0)_{i+1,j+1} = \begin{cases}2\sqrt{2i+1}\sqrt{2j+1}, & j-i \text{ is odd} \\ 0, & \text{otherwise}\end{cases}. \tag{22}$$

Now substituting Eq. (22) in Eq. (21), we get derivative operational matrix which is expressed as:

$$R_{i+1,j+1} = 2^n\begin{cases}2\sqrt{2i+1}\sqrt{2j+1}, & j-i \text{ is odd} \\ 0, & \text{otherwise}\end{cases}. \tag{23}$$

Transformation into ODE

In the present analysis, we use the derivative operational matrix, various properties of Legendre polynomial and the fluxes to transform Eq. (1) into a system of first-order ODEs.

Let us take any test function $v_h = \phi_{k,nl} \in V_{p,n}, k = 0, 1, 2..., p-1$. Now from first term of Eq. (14) and using Eq. (10), we have

$$\begin{aligned}\int_{I_{n,l}} \frac{\partial u_h}{\partial t}v_h\mathrm{d}x &= \int_{I_{n,l}} \frac{\partial(C_l^T(t)\Phi_l(x))}{\partial t}v_h\mathrm{d}x \\ &= \frac{\mathrm{d}C_l^T(t)}{\mathrm{d}t}\int_{I_{n,l}} \Phi_l(x)\phi_{k,nl}(x)\mathrm{d}x \\ &= \frac{\mathrm{d}c_{k,nl}}{\mathrm{d}t}.\end{aligned} \tag{24}$$

Taking the advantage of Eq. (23), we obtain the following calculation for second term of Eq. (14)

$$
\begin{aligned}
&\int_{I_{n,l}} \left(\alpha \, \frac{u_h^{\delta+1}}{\delta+1} - \frac{\partial u_h}{\partial x} \right) \frac{\partial \phi_{k,nl}(x)}{\partial x} \mathrm{d}x \\
&\quad = \alpha \int_{I_{n,l}} \frac{u_h^{\delta+1}}{\delta+1} \frac{\partial \phi_{k,nl}(x)}{\partial x} \mathrm{d}x - \int_{I_{n,l}} \frac{\partial u_h}{\partial x} \frac{\partial \phi_{k,nl}(x)}{\partial x} \mathrm{d}x \\
&\quad = \alpha \int_{I_{n,l}} \frac{u_h^{\delta+1}}{\delta+1} \frac{\partial \phi_{k,nl}(x)}{\partial x} \mathrm{d}x - \int_{I_{n,l}} \frac{\partial (C_l^T(t)\Phi_l(x))}{\partial x} \frac{\partial \phi_{k,nl}(x)}{\partial x} \mathrm{d}x \\
&\quad = \alpha \int_{I_{n,l}} \frac{u_h^{\delta+1}}{\delta+1} \frac{\partial \phi_{k,nl}(x)}{\partial x} \mathrm{d}x - C_l^T(t) \int_{I_{n,l}} \frac{\mathrm{d}\Phi_l(x)}{\mathrm{d}x} \frac{\mathrm{d}\phi_{k,nl}}{\mathrm{d}x} \mathrm{d}x \\
&\quad = \alpha \int_{I_{n,l}} \frac{u_h^{\delta+1}}{\delta+1} \frac{\partial \phi_{k,nl}(x)}{\partial x} \mathrm{d}x - 2^n C_l^T(t) r_0^T \int_{I_{n,l}} \Phi_l(x) \frac{\mathrm{d}\phi_{k,nl}}{\mathrm{d}x} \mathrm{d}x \\
&\quad = \alpha \int_{I_{n,l}} \frac{u_h^{\delta+1}}{\delta+1} \frac{\partial \phi_{k,nl}(x)}{\partial x} \mathrm{d}x - 4^n C_l^T(t) r_0^T r_0(:, k+1),
\end{aligned}
\tag{25}
$$

where $C_l^T(t)$ are the coefficients of the numerical solution u_h on a certain subinterval $I_{n,l}$ and $r_0(:, k+1)$ denotes the $(k+1)$th column of the derivative operational matrix r_0.

Now we need to calculate the convection and diffusion fluxes, i.e., third and fourth term of Eq. (14). For the computation of fluxes, we must first calculate u^+ and u^- at nodes x_l using the properties of Legendre polynomials

$$
P_k(1) = 1, \quad P_k(-1) = (-1)^k, \tag{26}
$$

and the Legendre basis functions

$$
\phi_k(0) = (-1)^k \sqrt{2k+1}, \quad \phi_k(1) = \sqrt{2k+1}. \tag{27}
$$

Thus for $l = 1, 2, 3, \dots, 2^n - 2$, we have

$$
\begin{aligned}
u^+(x_l) = u^+(x_{l+1}) &= C_l^T(t)\Phi_l(x_{l+1} = (l+1)/2^n) \\
&= 2^{n/2} C_l^T(t)[1, -\sqrt{3}, ..., (-1)^k \sqrt{2k+1} ..., \\
&\quad (-1)^{p-1}\sqrt{2p-1}] \\
&= 2^{n/2} C_l^T(t)\Phi_{-k},
\end{aligned}
\tag{28}
$$

where$\Phi_{-k} = [1, -\sqrt{3}, ..., (-1)^k \sqrt{2k+1} ..., (-1)^{p-1}\sqrt{2p-1}]$. Similarly we obtain

$$
\begin{aligned}
u^-(x_l) = u^-(x_{l+1}) &= 2^{n/2} C_l^T(t)[1, -\sqrt{3}, ..., \sqrt{2k+1} ..., \sqrt{2p-1}] \\
&= 2^{n/2} C_l^T(t)\Phi_k.
\end{aligned}
\tag{29}
$$

Additionally, for $l = 0$ and $l = 2^n - 1$, the boundary conditions are substituted into these computations.

Using Eqs. (28) and (29), we have [3]

$$
\left.\frac{\partial u_h^+}{\partial x}\right|_{x_l} = \left.\frac{\partial u_h^+}{\partial x}\right|_{x_{l+1}} = 2^{3n/2} C_l^T(t) r_0^T \Phi_{-k}, \tag{30}
$$

$$
\left.\frac{\partial u_h^-}{\partial x}\right|_{x_l} = \left.\frac{\partial u_h^-}{\partial x}\right|_{x_{l+1}} = 2^{3n/2} C_l^T(t) r_0^T \Phi_k, \tag{31}
$$

$$
\left.\frac{\partial^2 u_h^+}{\partial x^2}\right|_{x_l} = \left.\frac{\partial^2 u_h^+}{\partial x^2}\right|_{x_{l+1}} = 2^{5n/2} C_l^T(t) (r_0^T)^2 \Phi_{-k}, \tag{32}
$$

$$
\left.\frac{\partial^2 u_h^-}{\partial x^2}\right|_{x_l} = \left.\frac{\partial^2 u_h^-}{\partial x^2}\right|_{x_{l+1}} = 2^{5n/2} C_l^T(t) (r_0^T)^2 \Phi_k. \tag{33}
$$

Now using Eqs. (28), (29), (30), (31), (32) and (33), we can compute the value of fluxes given in third and fourth term of Eq. (14).

The third term of Eq. (14) becomes

$$
\begin{aligned}
&(\hat{f})_{x_{l+1}} v_{h(l+1)}^- - (\hat{f})_{x_l} v_{h(l)}^+ \\
&\quad = (\hat{f})_{x_l} (v_{h(l+1)}^- - v_{h(l)}^+) \\
&\quad = (\hat{f})_{x_l} (1 - (-1)^k)\sqrt{2k+1} \\
&\quad = \frac{\alpha}{2} \left[\frac{(2^{n/2} C_l^T(t)\Phi_{-k})^{\delta+1}}{\delta+1} + \frac{(2^{n/2} C_l^T(t)\Phi_k)^{\delta+1}}{\delta+1} \right. \\
&\qquad \left. - 2^{n/2} C_l^T(t)(\Phi_{-k} - \Phi k) \right] (1 - (-1)^k)\sqrt{2k+1} = A(t).
\end{aligned}
\tag{34}
$$

Next, the fourth term of Eq. (14) is calculated as:

$$
\begin{aligned}
&(\hat{q})_{x_{l+1}} v_{h(l+1)}^- - (\hat{q})_{x_l} v_{h(l)}^+ \\
&\quad = (\hat{q})_{x_l} (v_{h(l+1)}^- - v_{h(l)}^+) \\
&\quad = (\hat{q})_{x_l} (1 - (-1)^k)\sqrt{2k+1} \\
&\quad = 2^{3n/2} C_l^T \left[\beta_0(\Phi_{-k} - \Phi_k) + \frac{r_0^T}{2}(\Phi_{-k} + \Phi_k) \right. \\
&\qquad \left. + \beta_1 (r_0^T)^2 (\Phi_{-k} + \Phi_k) \right] ((-1)^k - 1)\sqrt{2k+1} = B(t).
\end{aligned}
\tag{35}
$$

Finally, the fifth term of Eq. (14) is calculated as:

$$
\begin{aligned}
\beta \int_{I_{n,l}} u_h(1 - u_h^\delta)\phi_{k,nl} \mathrm{d}x &= \beta \int_{I_{n,l}} u_h \phi_{k,nl} - \beta \int_{I_{n,l}} u_h^{\delta+1} \phi_{k,nl} \mathrm{d}x \\
&= \beta \int_{I_{n,l}} (C_l^T(t)\Phi_l(x))\phi_{k,nl} - \beta \int_{I_{n,l}} u_h^{\delta+1} \phi_{k,nl} \mathrm{d}x \\
&= \beta c_{k,nl} - \beta \int_{I_{n,l}} u_h^{\delta+1} \phi_{k,nl} \mathrm{d}x.
\end{aligned}
\tag{36}
$$

Now, we use Eqs. (24), (25), (34), (35) and (36) and obtain ODE system from the DLWG space discretization. For each k and l, where $k = 0, 1, 2, \dots, p-1, l = 0, 1, \dots, 2^n - 1$, we have

$$\begin{aligned}\frac{\mathrm{d}c_{k,nl}}{\mathrm{d}t} =&\alpha \int_{I_{n,l}} \frac{u_h^{\delta+1}}{\delta+1} \frac{\phi_{k,nl}(x)}{\mathrm{d}x}\mathrm{d}x - 4^n C_l^T(t) r_0^T r_0(:, k+1) - A(t)\\ &- B(t) + \beta c_{k,nl} - \beta \int_{I_{n,l}} u_h^{\delta+1} \phi_{k,nl} \mathrm{d}x.\end{aligned} \tag{37}$$

In addition, initial condition for the above system is given by

$$\begin{aligned}\int_{I_{n,l}} u(x,0)\phi_{k,nl}(x) &= \int_{I_{n,l}} u_h(x,0)\phi_{k,nl}(x)\\ &= \int_{I_{n,l}} C_l^T(0)\Phi_l(x)\phi_{k,nl}(x)\\ &= C_l^T(0) \int_{I_{n,l}} \Phi_l(x)\phi_{k,nl}(x)\\ &= c_{k,nl}(0).\end{aligned} \tag{38}$$

Equation (37) gives the first-order ODE system in the present DLWG method for $k = 0, 1, \dots, p-1$ and $l = 0, 1, \dots, 2^n - 1$; where $c_{k,nl}(0)$ are the coefficients of the initial numerical solution given by Eq. (38).

Numerical experiment

The gBFE (1) with initial condition (2) is transformed to the system of ODE (37), along with initial condition Eq. (38). Now we solve Eqs. (37) and (38) by using MATLAB for different values of α, β and δ to exhibit the efficiency and appropriateness of the DLWG method.

Tables 1 and 2 show the results of numerical solutions along with exact solutions and the corresponding absolute errors. Table 3 shows L_∞ error for $0 \le t \le 20$ and $0 \le x \le 1$ and Table 4 shows L_∞ error for $0 \le t \le 10$ and $0 \le x \le 1$. Accuracy of DLWG method is directly proportional to the value of n and p which is clearly illustrated through Tables 3 and 4. Figures 1 and 2 depict the comparisons between exact solution with DLWG solution and also corresponding error distributions have been presented respectively. Also 3-D surface solutions of modified Burgers–Fisher equation have been displayed in Figs. 3 and 4 respectively.

Conclusion

The prominent advantage of the proposed method is that the differential operator, boundary conditions and numerical fluxes involved in the elementwise computation can be done with less computational overhead. The operational matrix representation of the differential operator is simpler because of the advantages of the Legendre wavelets.

In the present article, the DLWG method is used to find the numerical solution of gBFE (1) and initial condition (2). The DLWG method transforms (1) into the system of first-order ODE given in Eq. (37) and the initial condition converted into Eq. (38). It is quite plausible that the numerical results rendered by the proposed method are quite satisfactorily agree with the exact solutions. The numerical experiments presented in this paper well establish the efficiency and applicability

Table 1 Comparison by absolute errors between exact and numerical solutions of Eq. (1) for $\alpha = 0.1, \beta = 0.1, \delta = 1$ and $t = 0.1$ with $n = 4$ and $p = 4$

$n = 4, p = 4$			
x	u (exact solution)	u (numerical solution)	Absolute error
0	0.502562477565014	0.502199371949117	0.000363105615897474
0.1	0.501312496985360	0.501069265586807	0.000243231398552646
0.2	0.500062499999674	0.499940006261988	0.000122493737686469
0.3	0.498812502232742	0.498810047166066	0.000002455066676288
0.4	0.497562519309387	0.497679404011903	0.000116884702515918
0.5	0.496312566853689	0.495952085329248	0.000360481524440459
0.6	0.495062660488206	0.494821017759040	0.000241642729165714
0.7	0.493812815833192	0.493690859854303	0.000121955978888866
0.8	0.492563048505816	0.492560077971703	0.000002970534113134
0.9	0.491313374119380	0.491428687554499	0.000115313435119724
1	0.490063808282541	0.490298242874817	0.000234434592275679

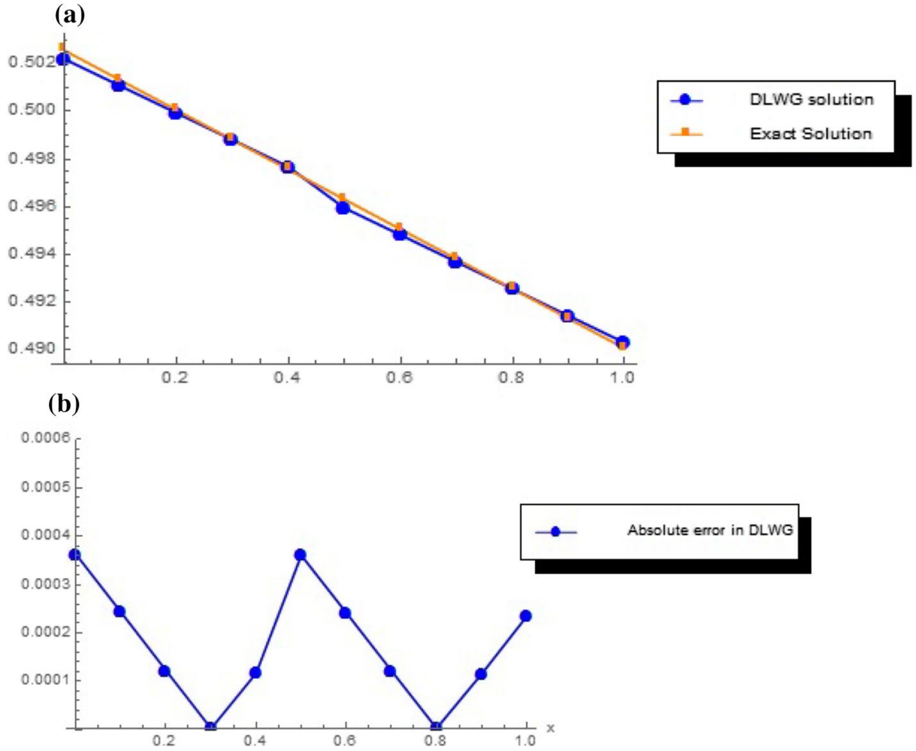

Fig. 1 **a** 2-D plot of exact and numerical solution of Eq. (1) for $\alpha = 0.1, \beta = 0.1, \delta = 1$ and $t = 0.1$ with $n = 4$ and $p = 4$, **b** absolute error distribution for DLWG solution

Table 2 Comparison by absolute errors between exact and numerical solutions of Eq. (1) for $\alpha = 0.01, \beta = 0.01, \delta = 1$ and $t = 0.1$ with $n = 4$ and $p = 4$

$n = 4, p = 4$			
x	u (exact solution)	u (numerical solution)	Absolute error
0	0.500250624979010	0.500221940401380	0.0000286845776303890
0.1	0.500125624997357	0.500106304374456	0.0000193206229005538
0.2	0.500000625000000	0.499994242564891	0.0000167220571147264
0.3	0.499875625002565	0.499881256503027	0.0000063824351093089
0.4	0.499750625020677	0.499767347077792	0.0000056315004620666
0.5	0.499625625069962	0.499600097726443	0.0000255273435186276
0.6	0.499500625166042	0.499480257469900	0.0000203676961426091
0.7	0.499375625324545	0.499369644882245	0.0000059804423004350
0.8	0.499250625561094	0.499256648378716	0.0000060228176220201
0.9	0.499125625891315	0.499141270635082	0.0000156447437667517
1	0.499000626330833	0.499035128187991	0.0000345018571586975

of the proposed technique. In brief summary, the proposed DLWG scheme maintains the advantages of both the wavelet Galerkin method and the DG method, such as the sparse representations of operators, consistency, higher-order accuracy and thus significantly improves the traditional DG method. In future, the proposed method may be implemented to solve the space-time fractional advection–diffusion equation, the space-time fractional advection–diffusion equation, the space fractional-order diffusion equation, time-fractional fourth-order reaction diffusion model, time fractional Tricomi-type model and time fractional cable model [9–14].

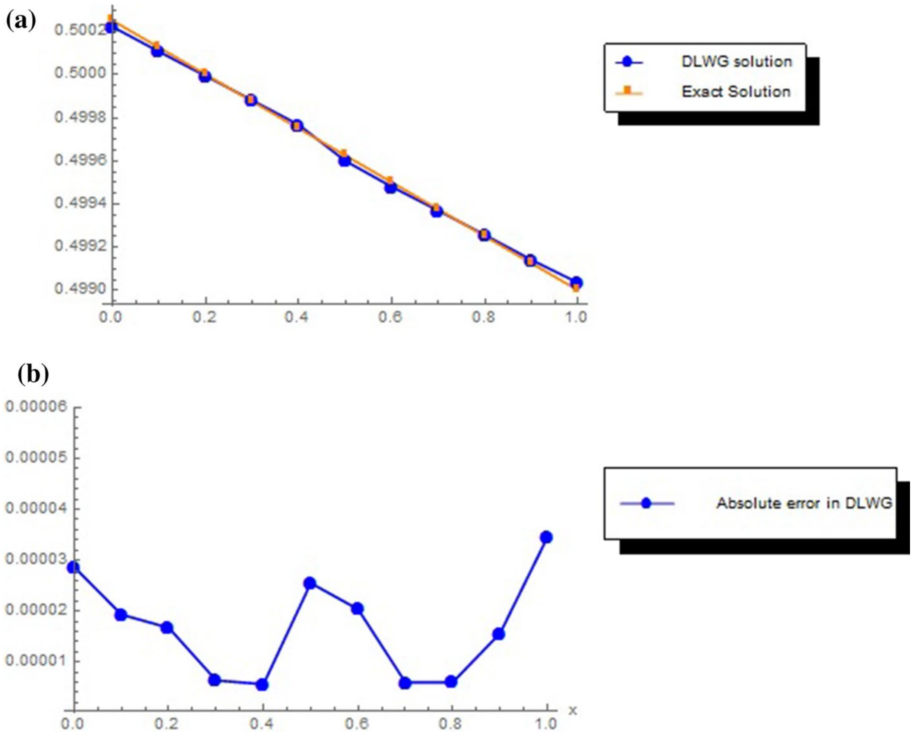

Fig. 2 **a** 2-D plot of exact and numerical solution of Eq. (1) for $\alpha = 0.01, \beta = 0.01, \delta = 1$ and $t = 0.1$ with $n = 4$ and $p = 4$, **b** absolute error distribution for DLWG solution

Table 3 L_∞ error for $\alpha = \beta = 0.1$ and $\delta = 2$ at $t = 20$

p	n	L_∞ error
2	3	$0.89064893 \times 10^{-2}$
3	3	0.9734468×10^{-2}
3	4	$0.50128079 \times 10^{-2}$
4	4	$0.47411297 \times 10^{-3}$
4	5	$0.18512906 \times 10^{-3}$
5	5	$0.89367901 \times 10^{-5}$

Table 4 L_∞ error for $\alpha = \beta = 0.5$ and $\delta = 2$ at $t = 10$

p	n	L_∞ error
2	3	$0.89064893 \times 10^{-2}$
3	3	$0.86729231 \times 10^{-3}$
3	4	$0.43017694 \times 10^{-3}$
4	4	$0.79540023 \times 10^{-4}$
4	5	$0.46313378 \times 10^{-4}$
5	5	$0.83479027 \times 10^{-6}$

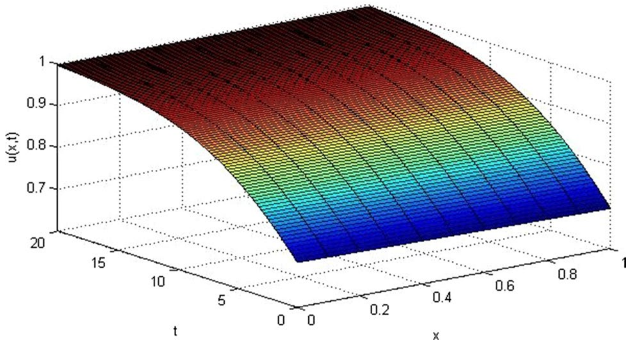

Fig. 3 3-D surface solution for $\alpha = 0.1, \beta = 0.1, \delta = 2$ and $0 \le t \le 20$ with $p = 5$ and $n = 5$

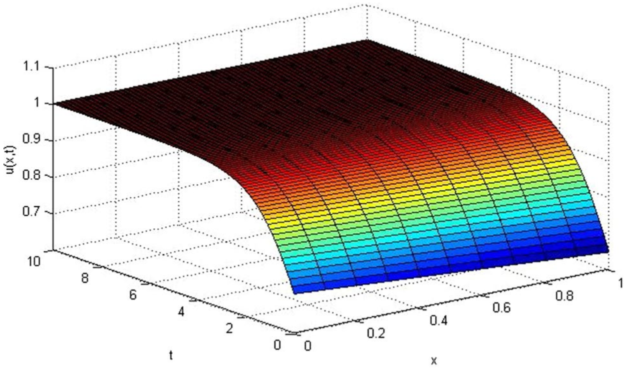

Fig. 4 3-D surface solution for $\alpha = \beta = 0.5$, $\delta = 2$ and $0 \le t \le 10$ with $p = 5$ and $n = 5$

Acknowledgements It is not out of place to mention that the author is deeply indebted to respected eminent professor Dr. A. M. Wazwaz of Saint Xavier University, Chicago, USA, for sparing his valuable time in reading the paper carefully for refinement of literature as well as grammatical corrections.

References

1. Chui, C.K.: Wavelets, A Mathematical Tool for Signal Analysis. SIAM Monographs on Mathematical Modeling and Computation. SIAM, Pennsylvania (1997)
2. Daubechies, I.: Ten Lectures on Wavelets. Society For Industrial and Applied Mathematics, Philadelphia (1992)
3. Zheng, X., Wei, Z.: Discontinuous Legendre wavelet element method for reaction-diffusion equation from mathematical chemistry. Int. J. Comput. Methods **16**(7), 1850113 (2019)
4. Zhang, R.-P., Zhang, L.-W.: Direct discontinuous Galerkin method for the generalized Burgers' Fisher equation. Chin. Phys. B **22**(9), 090206 (2012)
5. Malik, S.A., Qureshi, I.M., Amir, M., Malik, A.N., Haq, I.: Numerical solution to generalized Burgers'–Fisher equation using exp-function method hybridized with heuristic computation. PLoS ONE **10**(3), 1–15 (2015)
6. Sahu, P.K., Saha Ray, S.: Two-dimensional Legendre wavelet method for the numerical solutions of fuzzy integro-differential equations. J. Intell. Fuzzy Syst. **28**(3), 1271–1279 (2015)
7. Sahu, P.K., Saha Ray, S.: Legendre wavelets operational method for the numerical solutions of nonlinear Volterra integro-differential equations system. Appl. Math. Comput. **256**, 715–723 (2015)
8. Saha Ray, S., Gupta, A.K.: Numerical solution of fractional partial differential equation of parabolic type with dirichlet boundary conditions using two-dimensional Legendre wavelets method. J. Comput. Nonlinear Dyn. **11**(1), 011012 (2015)
9. Safdari, H., Aghdam, Y.E., Gómez-Aguilar, J.F.: Shifted Chebyshev collocation of the fourth kind with convergence analysis for the space–time fractional advection–diffusion equation. Eng. Comput. (2020). https://doi.org/10.1007/s00366-020-01092-x
10. Aghdam, Y.E., Mesgrani, H., Javidi, M., Nikan, O.: A computational approach for the space–time fractional advection–diffusion equation arising in contaminant transport through porous media. Eng. Comput. (2020). https://doi.org/10.1007/s00366-020-01021-y
11. Safdari, H., Mesgarani, H., Javidi, M., Aghdam, Y.E.: Convergence analysis of the space fractional-order diffusion equation based on the compact finite difference scheme. Comput. Appl. Math. (2020). https://doi.org/10.1007/s40314-020-1078-z
12. Nikan, O., Machado, J.A.T., Avazzadeh, Z., Jafari, H.: Numerical evaluation of fractional Tricomi-type model arising from physical problems of gas dynamics. J. Adv. Res. **25**, 205–216 (2020). https://doi.org/10.1016/j.jare.2020.06.018
13. Nikan, O., Machado, J.A.T., Golbabai, A.: Numerical solution of time-fractional fourth-order reaction–diffusion model arising in composite environments. Appl. Math. Model. **89**(1), 819–836 (2021). https://doi.org/10.1016/j.apm.2020.07.021
14. Nikan, O., Golbabai, A., Machado, J.A.T., Nikazad, T.: Numerical approximation of the time fractional cable model arising in neuronal dynamics. Eng. Comput. (2020). https://doi.org/10.1007/s00366-020-01033-8