Applied Mathematics, 2015, 6, 1581-1591
Published Online August 2015 in SciRes. http://www.scirp.org/journal/am
http://dx.doi.org/10.4236/am.2015.69141

Discontinuous Legendre Wavelet Galerkin Method for One-Dimensional Advection-Diffusion Equation

Xiaoyang Zheng, Zhengyuan Wei

College of Mathematics and Statistics, Chongqing University of Technology, Chongqing, China
Email: zhengxiaoyang@cqut.edu.cn

Received 29 July 2015; accepted 18 August 2015; published 21 August 2015

Abstract

This paper presents discontinuous Legendre wavelet Galerkin (DLWG) approach for solving one-dimensional advection-diffusion equation (ADE). Variational formulation of this type equation and corresponding numerical fluxes are devised by utilizing the advantages of both the Legendre wavelet bases and discontinuous Galerkin (DG) method. The distinctive features of the proposed method are its simple applicability for a variety of boundary conditions and able to effectively approximate the solution of PDEs with less storage space and execution. The results of a numerical experiment are provided to verify the efficiency of the designed new technique.

Keywords

Advection-Diffusion Equation, Legendre Wavelet, Discontinuous Galerkin Method, Discontinuous Legendre Wavelet Galerkin Method

1. Introduction

The advection-diffusion equation arises in many important applications, such as fluid dynamics, heat transfer and mass transfer etc. [1]-[9]. In this paper, we shall consider the one-dimensional convection-diffusion equation, which takes the form

$$\frac{\partial u}{\partial t}+a\frac{\partial u}{\partial x}=\mu\frac{\partial^2 u}{\partial x^2},\quad 0\le x\le 1, 0\le t\le T\ , \tag{1.1}$$

where a and μ are the speed of advection and diffusion coefficients respectively, and the function u is unknown. The initial and boundary conditions are

How to cite this paper: Zheng, X.Y. and Wei, Z.Y. (2015) Discontinuous Legendre Wavelet Galerkin Method for One-Dimensional Advection-Diffusion Equation. *Applied Mathematics*, **6**, 1581-1591. http://dx.doi.org/10.4236/am.2015.69141

$$u(x,0)=u_0(x), \tag{1.2}$$

and the boundary conditions satisfy

$$u(0,t)=g(t),\ \ u(1,t)=h(t), \tag{1.3}$$

where u_0, g and h are known functions.

There has been little progress in obtaining analytical solution to the advection-diffusion equation when initial and boundary conditions are complicated, even with constant coefficients a and v [7]. This is the reason why numerical solution of Equation (1.1) is very important. It is pointed out that a lot of numerical techniques for Equation (1.1) are by now well developed such as finite differences, finite elements, spectral procedures, Wavelet-Galerkin (WG) methods, DG methods and Graphical methods etc. [9]-[25]. Among these approaches, we need to emphasize on Wavelet Galerkin method, especially, Legendre Wavelet Galerkin technique and DG approach for solving partial differential equations (PDEs) because these methods are applied to constructing the DLWG approach proposed in this article.

Firstly, the DG method has emerged as an attractive tool for simulating the convection-diffusion problem [9]-[14]. The main advantage of the DG method lies in its accuracy and flexibility thanks to its high degree of locality. Secondly, the reason for such fast development of the WG approach may be the fact that many nonlinear PDEs have solutions containing local phenomena and interactions among several scales, which can be well-represented in wavelet bases owing to their nice properties, such as compact support and vanishing moment [15]-[22]. However, their main limitations are the difficulties to adapt them to non-periodic geometries and to append specific boundary conditions. Thirdly, what most interests us is that the Legendre wavelet approach is widely implemented to solve PDEs because of its rich properties, for example, expressions in closed form, orthogonality, compact support and vanishing moments [14]-[25].

In this paper, the DLWG technique is constructed by borrowing the idea of the DG method. The DLWG approach is based on the variational formulation for the solution of Equation (1.1) and takes advantage of an elementwise discontinuous Legendre wavelet approximation, where numerical information only communicates locally via numerical fluxes, to cope with complicated geometries and to represent the dynamics and structure of highly complex solutions. Especially, compared with the LWG method, the discontinuity of Legendre wavelet functions at interfaces of element to element and the boundary conditions are easy to be solved by using the numerical fluxes. Furthermore, the rich properties of Legendre wavelet bases and the use of the discontinuous elements can produce block-diagonal, sparse and lower dimensional mass matrices, which can be easily inverted by hand and stored efficiently in computer-memory compared with the WG method. Of course, the stability and approximate error of the DLWG approach are also addressed in this article. Finally, the DLWG approach utilizes the discontinuous feature at nodes of the Legendre wavelet bases combined with discontinuous finite elements to discretize the space variable and the spacial derivatives to produce a system of first-order ODEs in time for Equation (1.1). We solve this system by using the TVD Runge-Kutta method [11], and obtain good numerical results, illustrating that this scheme is very simple and computationally efficient.

This paper is organized as follows: in Section 2, descriptions of the Legendre wavelet and its rich properties are given. Section 3 obtains the variational form of Equation (1.1) by the DLWG method. Section 4 derives the computations of the derivative operator and the numerical fluxes. Through these calculations, Equation (1.1) is transformed into an ODE system with time. Section 5 addresses the stability analysis of the DLWG approach. In Section 6, the results of a numerical experiment are presented to demonstrate the efficiency of the DLWG method. Conclusions of the proposed method and some suggestions for future research are given at the end of Section 7.

2. Legendre Wavelet

In this section, we briefly review the Legendre wavelet bases, and introduce our notations and some auxiliary results that will be used later [22]-[24]. Let $L_k(x)$ denote the Legendre polynomial of degree k, which is inductively defined as follows:

$$\begin{aligned} &L_0(x)=1,\ \ L_1(x)=x,\\ &L_{k+2}(x)=\frac{2k+3}{k+2}xL_{k+1}(x)-\frac{k+1}{k+2}L_k(x). \end{aligned} \tag{2.1}$$

Let $\phi_k(x)$ denote the Legendre scale function defined as

$$\phi_k(x)=\begin{cases}\sqrt{2k+1}L_k(2x-1), & x\in[0,1],\\ 0, & x\notin[0,1].\end{cases} \tag{2.2}$$

For $n=0,1,2,\cdots$, and $l=0,1,2,\cdots,2^n-1$, define the interval $I_{nl}=\left[2^{-n}l,2^{-n}(l+1)\right)$. For $p=1,2,\cdots$, define $V_{p,n}$ as a subspace of piecewise polynomial functions, $V_{p,n}=\{f:f|_{I_{nl}}$ is a polynomial of degree strictly less than *p*; and *f* vanishes elsewhere}.

The whole set $\{\phi_k\}_{k=0}^{p-1}$ forms an orthonormal basis for $V_{p,0}$. Generally, the subspace $V_{p,n}$ is spanned by $2^n p$ functions which are obtained from $\phi_0,\phi_1,\cdots,\phi_{p-1}$ by dilations and translations, *i.e.*,

$$V_{p,n}:=V_{p,nl}\ \text{spa}\ \left\{\phi_{k,nl}(x)=2^{n/2}\phi_k\left(2^n x-l\right),0\le k\le p-1,0\le l\le 2^n-1\right\}$$

which forms an orthonormal basis and

$$V_{p,0}\subset V_{p,1}\subset\cdots\subset V_{p,n}\subset\cdots. \tag{2.3}$$

In order to intuitively understand the Legendre scale functions, we let the scale level $n=2$ and $p=3$, respectively. **Figure 1** plots the scale functions.

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

$$\mathrm{P}_n f(x)=\sum_{l=0}^{2^n-1}\sum_{k=0}^{p-1}s_{k,nl}\phi_{k,nl}(x), \tag{2.4}$$

where P_n is the finest scale projection and $s_{k,nl}$ are scale coefficients. Furthermore, the approximate estimation satisfies

$$\left\|f-\sum_{l=0}^{2^n-1}\sum_{k=0}^{p-1}s_{k,nl}\phi_{k,nl}(x)\right\|_2\sim O\left(2^{-np}\right), \tag{2.5}$$

which demonstrates the approximation error exponentially convergences with the level *n* of resolution and the order *p* of the Legendre wavelet bases [22]. The above nice properties demonstrate that the Legendre wavelet bases can be very efficiently applied to the numerical solution of Equation (1).

3. Discontinuous Legendre Wavelet Galerkin Variational Form

In this section, we derive the weak formulation of Equation (1.1) by the DLWG technique in detail. The computational domain $[0,1]$ is firstly divided into $N=2^n$ elements, *i.e.*, subintervals I_{nl} as described in Section 2,

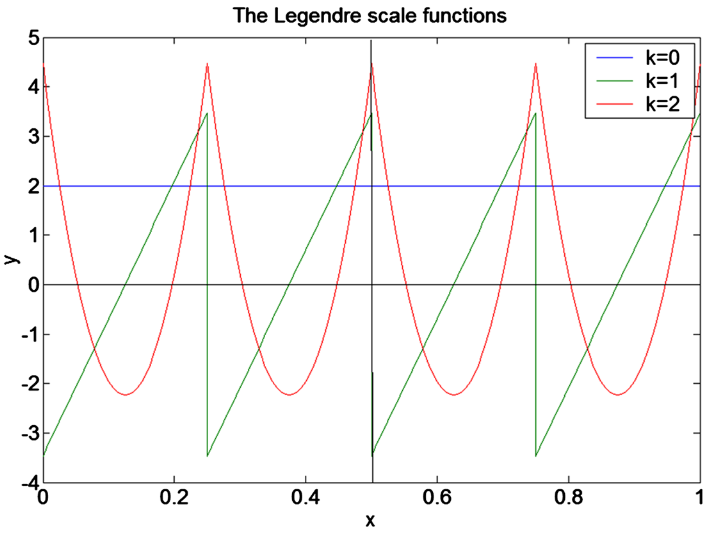

Figure 1. Legendre scale functions with $n=2$ and $p=3$.

so the size of each element is $h = 2^{-n}$. Secondly, the spaces of the approximation solutions and the test functions are defined as the Legendre wavelet functions spaces $V_{p,hl} := V_{p,nl}$ introduced in Section 2. Thirdly, for a specific l, the numerical solution for $x \in I_{nl}$ in space $V_{p,hl}$ can be approximated by

$$\mathrm{P}_n u(x,t) = \sum_{k=0}^{p-1} c_{k,nl}(t)\phi_{k,nl}(x) = C_{hl}^{\mathrm{T}}(t)\Phi_l(x), \tag{3.1}$$

where $C_{hl}(t) = C_l(t) = \left[c_{0,nl}(t), c_{1,nl}(t), \cdots, c_{p-1,nl}(t)\right]^{\mathrm{T}}$ is unknown time-dependent quantity to be determined from the initial conditions and the weak solution form of Equation (1.1), and

$$\Phi_l(x) = \left[\phi_{0,nl}(x), \phi_{0,nl}(x), \cdots, \phi_{p-1,nl}(x)\right]^{\mathrm{T}}$$

is the vector of the Legendre wavelet bases at the finest decomposition level *n*.

Now we utilize u_l^+ and u_l^- to denote the values of u at $x_l = 2^{-n} l$, $l = 1, 2, \cdots, 2^n - 1$ from right and left, respectively,

$$u_l^+ = \lim_{\varepsilon \to 0+} u(x_l + \varepsilon),\ u_l^- = \lim_{\varepsilon \to 0+} u(x_l - \varepsilon).$$

We also let the usual notations $\{u\} = \left(u^+ + u^-\right)/2$ and $[[u]] = u^+ - u^-$ represent the mean and the jump of function *u* at each element I_{nl} boundary point, respectively. Then the semi-discrete *i.e.*, space discretization, the DLWG approach is applied to Equation (1.1) and the corresponding initial and the boundary conditions,

$$u(x,0) = u_0(x),\ u(0,t) = g(t),\ u(1,t) = h(t). \tag{3.2}$$

In order to determine the approximate solution $u_h \in V_{p,hl}$, we implement a weak formulation to multiply Equation (1.1) and (3.2) by all test functions, that is to say, Legendre wavelet functions for any $v_h \in V_{p,hl}$ and integrate over each element I_{nl}, and then obtain the following

$$\int_{I_{nl}} \frac{\partial u_h}{\partial t} v_h \mathrm{d}x + a\int_{I_{nl}} \frac{\partial u_h}{\partial x} v_h \mathrm{d}x = \mu \int_{I_{nl}} \frac{\partial^2 u_h}{\partial x^2} v_h \mathrm{d}x, \tag{3.3a}$$

$$\int_{I_{nl}} u_h(x,0) v_h \mathrm{d}x = \int_{I_{nl}} u_0(x) v_h \mathrm{d}x, \tag{3.3b}$$

where $l = 0, 1, 2, \cdots, 2^n - 1$. After a simple formal integration by parts over Equation (3.3), we have

$$\int_{I_{nl}} \frac{\partial u_h}{\partial t} v_h \mathrm{d}x - \int_{I_{nl}} \left(a u_h - \mu \frac{\partial u_h}{\partial x}\right) \frac{\partial v_h}{\partial x} \mathrm{d}x = \left(\overline{a u_h} - \overline{q}\right)_{x_l} v_{hl}^+ - \left(\overline{a u_h} - \overline{q}\right)_{x_{l+1}} v_{h(l+1)}^- \tag{3.4}$$

for each element I_{nl}. The functions $\overline{a u_h}$ and $\overline{q}$ in Equation (3.4) are convection and diffusion numerical fluxes, respectively, which are single-valued functions defined at the element interfaces and in general depend on the values of numerical solution u_h or its derivatives from both sides of the interfaces. Since the function u_h is discontinuous at the points x_l, we must also replace the nonlinear convection and diffusion fluxes by the numerical fluxes $\overline{a u_h}$ and $\overline{q}$, which appear from integration by parts. A suitable choice for these numerical fluxes is the key ingredient for the stability of the DLWG scheme.

There are two types of numerical fluxes: one is the convection numerical flux $\overline{a u_h}$; the other is the diffusion numerical flux $\overline{q}$. In this work, the convection flux $\overline{a u_h}$ is chosen to be the local Lax-Friedrichs flux [11]

$$\overline{a u_h} = \frac{1}{2} a\left[(1-a) u_h^+ + (1+a) u_h^-\right]. \tag{3.5}$$

In addition, the diffusion flux $\overline{q}$ is chosen as [11]

$$\overline{q}_{x_l} = 2^n \beta_0 \mu [[u_h]] + \mu \left\{\frac{\partial u_h}{\partial x}\right\} + 2^{-n} \beta_1 \mu \left[\left[\frac{\partial^2 u_h}{\partial x^2}\right]\right]. \tag{3.6}$$

We note that the choice of numerical fluxes follows the same principle as those for the LDG method. However, numerical experience suggests that as the degree *k* of the approximate solution increases, the choice of the numerical fluxes does not have significant impact on the quality of the approximations [11].

4. Computation of the Variational Form

In this section, we concretely evaluate each term of Equation (3.4) obtained in Section 3 by taking advantage of the characteristics of the Legendre wavelet bases.

4.1. Calculating the Matrix of Derivatives

Since the Legendre basis functions are discontinuous at nodes x_l when the order of the bases is odd, representations of derivative operators do not exist in the usual sense. The proposed approach by Beylkin *et al.* [22] is based on defining weak representations of the derivative operator. The transition matrices have large dimensions $2^n p\times 2^n p$, although they have block diagonal structures. In this subsection, we can adopt another calculation technique of the differential operator, *i.e.*, computing on each element I_{nl} other than whole interval $[0,1]$. This approach consists of the DG method for solving the PDEs and avoids the discontinuity of the Legendre wavelet bases at interfaces. Furthermore, the interactions of the adjacent elements are jointed by using the convection and diffusion fluxes. As we shall show below, this representation of the derivative on each element is an important advantage of the lower dimension transition matrices $p\times p$.

We now let *D* denote the derivative operator and consider $D_{p,n}:V_{p,n}\to V_{p,n}$ (the projection $P_{p,n}DP_{p,n}$ of *D* on $V_{p,n}$) for some fixed resolution level *n*. Then let us consider solution *u* and $P_{p,n}u, D_{p,n}u\in V_{p,n}$ with expansions by the Legendre scale functions,

$$\left(P_{p,n}u\right)(x)=\sum_{m=0}^{2^n-1}\sum_{j=0}^{p-1}s_{j,nm}\phi_{j,nm}(x), \tag{4.1}$$

$$\left(D_{p,n}u\right)(x)=\sum_{l=0}^{2^n-1}\sum_{i=0}^{p-1}\overline{s_{i,nl}}\phi_{i,nl}(x). \tag{4.2}$$

Our goal is to find the $p\times p$ transition matrices r_{nlm} for $l,m=0,1,\cdots,2^n-1$, which satisfy

$$\overline{s_{i,nl}}=\sum_{m=0}^{2^n-1}\sum_{j=0}^{p-1}\left[r_{nlm}\right]_{ij}s_{j,nm}. \tag{4.3}$$

Then the coefficient $\left[r_{nlm}\right]_{ij}$ would necessarily be given by

$$\left[r_{l'}\right]_{ij}=\int_0^1\phi_i(x)D\phi_j(x+l')\mathrm{d}x\left[r_{nlm}\right]_{ij}=\int_{2^{-n}l}^{2^{-n}(l+1)}\phi_{i,nl}(x)D\phi_{j,nm}(x)\mathrm{d}\,x=2^n\left[r_{l-m}\right]_{ij}, \tag{4.4}$$

where

$$\left[r_{l'}\right]_{ij}=\int_0^1\phi_i(x)D\phi_j(x+l')\mathrm{d}x$$

is the representation of *D* on the coarsest scale $V_{p,0}$. Also, because we compute the transition matrices on each element I_{nl}, only the same element is involved, that is $l'=0$. Consequently, the matrix in (4.4) is replaced by

$$\left[r_0\right]_{ij}=\int_0^1\phi_i(x)\frac{\mathrm{d}}{\mathrm{d}x}\phi_j(x)\mathrm{d}x, \tag{4.5}$$

which again is a formal expression at this point, where $i,j=0,1,\cdots,p-1$ and r_0 is a $p\times p$ matrix. Additionally, the representation of transition matrix on the decomposition level *n* can be found on each element I_{nl} by rescaling

$$r_{n,0}=2^n r_0. \tag{4.6}$$

We now return to how to provide explicit calculation for each element of the matrix in (4.4) for the Legendre scale bases. Using a relation for the Legendre polynomials

$$(2j+1)L_j(x)=L'_{j+1}(x)-L'_{j-1}(x), \tag{4.7}$$

we obtain for the first derivative

$$\frac{\phi_j'(x)}{2\sqrt{2j+1}} = \sqrt{2j-1}\phi_{j-1}(x) + \sqrt{2j-5}\phi_{j-3}(x) + \cdots + \begin{cases} \phi_0(x), & j \text{ odd}, \\ \sqrt{3}\phi_1(x), & j \text{ even}. \end{cases} \tag{4.8}$$

Substituting (4.8) into (4.4), we find that the (*i* + 1, *j* + 1)-th element of the matrix $(r_0)_{i+1,j+1} = [r_0]_{ij}$ satisfies

$$(r_0)_{i+1,j+1} = 2\Gamma_{ij}v_{ij}, \tag{4.9}$$

where $\Gamma_{ij} = \sqrt{2i+1}\sqrt{2j+1}$ and

$$v_{ij} = \begin{cases} 1, & j-i = 1,3,5,\cdots, \\ 0, & \text{otherwise}. \end{cases} \tag{4.10}$$

For example, let *p* = 3, we can obtain

$$r_0 = 2\begin{pmatrix} 0 & \sqrt{3} & 0 \\ 0 & 0 & \sqrt{15} \\ 0 & 0 & 0 \end{pmatrix}. \tag{4.11}$$

The transition matrix $r_{n,0}$ derived above is applied to numerical solution of Equation (1.1). They can also be used to solve problems such as calculus of variations, differential equations, optimal control and integral equations.

4.2. Transformation PDE into ODE and Time Discretization

In this subsection, we use the matrix of differential operator and the fluxes proposed in above subsections to transform Equation (1.1) into a system of ODEs in time.

For any test function $\phi_{k,nl}(x) \in V_{p,nl}$, $k = 0,1,\cdots,p-1$, we firstly substitute (3.1) into the first term in (3.4) and then have

$$\int_{I_{nl}} \frac{\mathrm{d}C_l^{\mathrm{T}}}{\mathrm{d}t}\Phi_l(x)\phi_{k,nl}(x)\mathrm{d}x = \frac{\mathrm{d}c_{k,nl}(t)}{\mathrm{d}t}. \tag{4.12}$$

Secondly, taking advantage of the result of (4.6), we obtain the concrete calculation of the second term in (3.4) satisfying

$$\begin{aligned} \int_{I_{nl}} \left(au_h - \mu\frac{\partial u_h}{\partial x}\right)\frac{\partial \phi_{k,nl}(x)}{\partial x}\mathrm{d}x &= 2^n aC_l^{\mathrm{T}}(t)r_0(:,k+1) - \mu\int_{I_{nl}} \frac{\partial u_h}{\partial x}\cdot\frac{\partial \phi_{k,nl}(x)}{\partial x}\mathrm{d}x \\ &= a2^n C_l^{\mathrm{T}}(t)r_0(:,k+1) - \mu 4^n C_l^{\mathrm{T}}(t)r_0^{\mathrm{T}}r_0(:,k+1), \end{aligned} \tag{4.13}$$

where $C_l^{\mathrm{T}}(t)$ are the presentation coefficients of the numerical solution on a certain subinterval I_{nl}, and $r_0(:,k+1)$ denotes the (*k* + 1)-th column of the derivative transition matrix r_0.

Up to present, we need to compute the convection and diffusion fluxes, *i.e.*, the third and fourth terms in (3.4). According to the definitions of the fluxes (3.5) and (3.6), we must first evaluate the approximate values of u^+ and u^- with needed accuracy at notes x_l. We use the following properties of the Legendre polynomials

$$L_k(1) = 1,\ L_k(-1) = (-1)^k, \tag{4.14}$$

and obtain the corresponding results of the Legendre basis functions such that

$$\phi_k(0) = (-1)^k\sqrt{2k+1},\ \phi_k(1) = \sqrt{2k+1}. \tag{4.15}$$

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

$$\begin{aligned} u^+(x_l) &= u^+(x_{l+1}) = C_l^{\mathrm{T}}(t)\Phi_l\left(x_{l+1} = (l+1)/2^n\right) \\ &= 2^{n/2}C_l^{\mathrm{T}}(t)\left[1,-\sqrt{3},\cdots,(-1)^k\sqrt{2k+1},\cdots,(-1)^{p-1}\sqrt{2p-1}\right]^{\mathrm{T}} \\ &= 2^{n/2}C_l^{\mathrm{T}}(t)\Phi_{(-1)^k c}, \end{aligned} \tag{4.16}$$

where we let $\Phi_{(-1)^{k}c} = \left[1, -\sqrt{3}, \cdots, (-1)^{k}\sqrt{2k+1}, \cdots, (-1)^{p-1}\sqrt{2p-1}\right]^{\mathrm{T}}$. Similarly, we obtain

$$u^{-}(x_l) = u^{-}(x_{l+1}) = 2^{n/2} C_l^{\mathrm{T}}(t)\left[1, \sqrt{3}, \cdots, \sqrt{2k+1}, \cdots, \sqrt{2p-1}\right]^{\mathrm{T}} = 2^{n/2} C_l^{\mathrm{T}}(t)\Phi_c. \tag{4.17}$$

Additionally, for $l = 0$ and $l = 2^n - 1$, the boundary conditions (1.3) are subsituted into these computations. Using (4.16) and (4.17), we have

$$\begin{gathered} \left.\frac{\partial u_h^+}{\partial x}\right|_{x_l} = \left.\frac{\partial u_h^+}{\partial x}\right|_{x_{l+1}} = 2^{3n/2} C_l^{\mathrm{T}}(t) r_0^{\mathrm{T}} \Phi_{(-1)^{k}c}, \quad \left.\frac{\partial u_h^-}{\partial x}\right|_{x_l} = \left.\frac{\partial u_h^-}{\partial x}\right|_{x_{l+1}} = 2^{3n/2} C_l^{\mathrm{T}}(t) r_0^{\mathrm{T}} \Phi_c, \\ \left.\frac{\partial^2 u_h^+}{\partial x^2}\right|_{x_l} = \left.\frac{\partial^2 u_h^+}{\partial x^2}\right|_{x_{l+1}} = 2^{5n/2} C_l^{\mathrm{T}}(t)\left(r_0^{\mathrm{T}}\right)^2 \Phi_{(-1)^{k}c}, \end{gathered} \tag{4.18}$$

$$\left.\frac{\partial^2 u_h^-}{\partial x^2}\right|_{x_l} = \left.\frac{\partial^2 u_h^-}{\partial x^2}\right|_{x_{l+1}} = 2^{5n/2} C_l^{\mathrm{T}}(t)\left(r_0^{\mathrm{T}}\right)^2 \Phi_c. \tag{4.19}$$

Now, using (3.5) and (3.6), we can obtain the computations of the fluxes satisfying

$$\begin{aligned} &\overline{au_h}(x_l) v_{hl}^{+}(x_l) - \overline{au_h}(x_{l+1}) v_{h(l+1)}^{-}(x_{l+1}) = \overline{au_h}(x_l)\left[v_{hl}^{+}(x_l) - v_{h(l+1)}^{-}(x_{l+1})\right] \\ &= \left[\frac{1}{2}\left((1-a) 2^{n/2} C_l^{\mathrm{T}}(t)\Phi_{(-1)^{k}c} + (1+a) 2^{n/2} C_l^{\mathrm{T}}(t)\Phi_c\right)\right] \cdot \left[(-1)^k - 1\right]\sqrt{2k+1} \\ &= A\overline{C_l^{\mathrm{T}}(k,t)}, \end{aligned} \tag{4.20}$$

$$\begin{aligned} &\overline{q}_{x_{ll}} v_l^{+} - \overline{q}_{x_{l+1}} v_{l+1}^{-} = \overline{q}_{x_l}\left(v_l^{+} - v_{l+1}^{-}\right) = \overline{q}_{x_l} 2^{n/2}\left[(-1)^k - 1\right]\sqrt{2k+1} \\ &= 2^{3n/2}\mu\left[\beta_0 C_l^{\mathrm{T}}\left(\Phi_{(-1)^{k}c} - \Phi_c\right) + C_l^{\mathrm{T}} r_0^{\mathrm{T}}\left(\Phi_{(-1)^{k}c} + \Phi_c\right) + \beta_1 C_l^{\mathrm{T}}\left(r_0^{\mathrm{T}}\right)^2\left(\Phi_{(-1)^{k}c} - \Phi_c\right)\right]\left[1 - (-1)^k\right]\sqrt{2k+1} \\ &= 2^{3n/2} u\overline{C_l^{\mathrm{T}}}(k,t,r_0). \end{aligned} \tag{4.21}$$

With (4.20) and (4.21), we obtain

$$\left(\overline{au_h} - \overline{q}\right)_{x_l} v_{hl}^{+} - \left(\overline{au_h} - \overline{q}\right)_{x_{l+1}} v_{h(l+1)}^{-} = A\overline{C_l^{\mathrm{T}}(k,t)} - 2^{3n/2}\mu\overline{C_l^{\mathrm{T}}}(k,t,r_0). \tag{4.22}$$

Finally, we use (4.12), (4.13) and (4.22) to obtain the ODE systems from the DLWG space discretization. For each k and l, $k = 0, 1, \cdots, p-1$, $l = 0, 1, \cdots 2^n - 1$, we have

$$\frac{\mathrm{d}c_{k,nl}(t)}{\mathrm{d}t} + a 2^n C_l^{\mathrm{T}}(t) r_0(:, k+1) - \mu 4^n C_l^{\mathrm{T}}(t) r_0^{\mathrm{T}} r_0(:, k+1) = A\overline{C_l^{\mathrm{T}}(k,t)} - 2^{3n/2}\mu\overline{C_l^{\mathrm{T}}}(k,t,r_0). \tag{4.23}$$

In addition, the initial condition (1.2) is represented as

$$u_h(x,0) = \sum_{k=0}^{p-1} c_{k,nl}(0)\phi_{k,nl}, \tag{4.24}$$

where $c_{k,nl}(0)$ are the coefficients of the initial numerical solution. When l = 0 and $l = 2^n - 1$, each term in (4.23) can be computed by using the boundary conditions (1.3). We now rewrite the p ODE systems (4.23) to a short-handed ODE system of the form

$$\frac{\mathrm{d}U_h(t)}{\mathrm{d}t} = L(U_h, k, t) \text{ in } t \in (0, T). \tag{4.25}$$

where $U_h(t) = C_{hl}(t)$ is the vector of the coefficients of the numerical solution u_h on the subinterval I_{nl} of Equation (1.1) by using the DLWG method proposed in this paper.

In the current work, the ODE system (4.25) is discretized over time by using the total variation diminishing (TVD) high-order Runge-Kutta method as follows [11]

$$U_h^{(1)} = U_h^m + \Delta t L\left(U_h^m, t_m\right),$$

$$U_h^{(2)} = \frac{3}{4}U_h^m + \frac{1}{4}U_h^{(1)} + \frac{1}{4}\Delta t L\left(U_h^{(1)}, t_m + \Delta t\right),$$

$$U_h^{(m+1)} = \frac{1}{3}U_h^m + \frac{2}{3}U_h^{(2)} + \frac{2}{3}\Delta t L\left(U_h^{(2)}, t_m + \frac{1}{2}\Delta t\right), \tag{4.26}$$

where $m = 0,1,\cdots$ denotes the time discretization step, and Δt is the time step.

Solving (4.25) by the above time discretization approach, we can obtain the finest resolution scale coefficients $C_{hl}(t)$ on each subinterval I_{nl} in space $V_{p,n}$. Consequently, for all *l*, the numerical approximate solution to Equation (1.1) is obtained on the *n* resolution. In comparison with the WG method, we do have more equations to solve. However, the reason why our technique is valid is that we do not try to solve a large $2^n p$ system but solve effectively 2^n small *p* systems. This results in a dimensional reduction, which is quite appealing for *h*-*p* adaptivity. If the resolution level *n* is large enough, the lower dimensional matrices would remarkably reduce the storage and time cost complexity needed to solve the full system.

5. Stability Analysis

In this section, we shall address the stability property of the DLWG scheme we just proposed. For simplicity of discussion, we shall only consider the case of μ being positive from now on, and only consider periodic boundary conditions, which is the type of boundary conditions analyzed in this article. For general boundary conditions, the choice of the numerical fluxes should be adjusted at the boundary. See [11] for details.

Theorem 1. The numerical scheme (3.4) with the fluxes choices (3.5) and (3.6), respectively, is L_2 stable, *i.e.*

$$\left\|u_h(t)\right\| \le \left\|u_h(0)\right\|.$$

Proof. According to the Equation (1.1), we let $G(u) = au$ and $K(u) = \int G(u)\mathrm{d}u$. In (3.4), we let $v_h = u_h$ and sum over *l* to obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{1}{2}\int_I u_h^2 \mathrm{d}x + \frac{1}{2}\mu\int_I\left(\frac{\partial u_h}{\partial x}\right)^2 \mathrm{d}x + \sum_{l=0}^{2^n-1}\Theta_l = 0, \tag{5.1}$$

where $\Theta_l = \left(\left[\left[K(u_h)\right]\right] - \bar{G}\left[\left[u_h\right]\right] + \frac{1}{2}\bar{q}\left[\left[u_h\right]\right]\right)_l$, $\bar{G}$ is the numerical flux for the function $G(u)$. It is now easy to show that $\left[\left[K(u_h)\right]\right] - \bar{G}\left[\left[u_h\right]\right] \ge 0$ following the proof of the cell entropy inequality in [11], using the fact that $\bar{G}$ is a monotone flux. Consequently,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\frac{1}{2}\int_I u_h^2 \mathrm{d}x &= -\frac{1}{2}\mu\int_I\left(\frac{\partial u_h}{\partial x}\right)^2 \mathrm{d}x - \sum_{l=0}^{2^n-1}\Theta_l \\ &\le -\frac{1}{2}\mu\int_I\left(\frac{\partial u_h}{\partial x}\right)^2 \mathrm{d}x - \frac{1}{2}\sum_{l=0}^{2^n-1}\bar{q}\left[\left[u_h\right]\right],\end{aligned} \tag{5.2}$$

Choosing suitable parameters β_0, β_1 of the flux $\bar{q}$ in (3.6), following the same argument as the literature [11], we have

$$-\frac{1}{2}\sum_{l=0}^{2^n-1}\bar{q}\left[\left[u_h\right]\right] \le 0. \tag{5.3}$$

Thus, we can obtain in (5.2)

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{1}{2}\int_I u_h^2 \mathrm{d}x \le 0, \tag{5.4}$$

which finishes the proof.

6. Numerical Experiment

In this section, we provide numerical experiment for numerically solving Equation (1.1) to illustrate the performance of the DLWG approach.

The values of various used parameters are $a = 1\,\mathrm{m/s}$, $\nu = 0.01\,\mathrm{m}^2/\mathrm{s}$, respectively. This means that we consider the equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0.01\frac{\partial^2 u}{\partial x^2} \tag{6.1}$$

defined on $0 \le x \le 1$, $0 \le t \le 1$. The exact solution is taken by [7] and given as

$$u(x,t) = \frac{0.025}{\sqrt{0.000625 + 0.02t}}\exp\left[-\frac{(x+0.5-t)^2}{0.00125+0.04t}\right]. \tag{6.2}$$

The corresponding initial and boundary conditions are decided by (6.1). Because the exact solution of the Equation (6.2) is known, we can compute the error between numerical solution and exact solution.

We calculate the solution up to $t = 1$. The parameters of the numerical flux $\bar{q}$ in (3.6) are chosen as $\beta_0 = 4$, $\beta_1 = 1/11$. We list the computational results in **Table 1** and **Table 2** at time $t = 0.5$ and $t = 0.9$, respectively.

In the tables, Column 1 indicates the spatial order *k*, Column 2 shows the finest scale *n* used, and Column 3 contains the size of time steps Δt. In Column 4, we evaluate the L_∞ error of the solution on 100 × 100 points and comparing it with the exact solution in (6.2). Additionally, in order to describe the evolution at desired accuracy, **Figure 2** describes the numerical solution of Equation (6.1) at different time and on the decomposition level $n = 3$ and $p = 3$, respectively.

As shown in the numerical experiment, the numerical solution from the DLWG method is in good agreement with the exact one and illustrates the accuracy and capacity of the DLWG approach proposed in this article.

7. Conclusion

In this paper, the numerical method has been used to solve the advection-diffusion equation with specified initial and boundary conditions. The numerical experiment is presented to demonstrate the high order accuracy and validity of this technique. In particular, this method can be generalized to multi-dimensional cases and be applied to other kinds of PDEs and integro-differential equations.

Table 1. Results with different parameters at time *t* = 0.5.

p	n	Δt	L_∞ error
2	1	10^{-2}	3.0194×10^{-2}
	2	10^{-2}	4.8795×10^{-3}
3	1	10^{-2}	7.6437×10^{-3}
	2	10^{-2}	1.3529×10^{-5}

Table 2. Results with different parameters at time *t* = 0.9.

p	n	Δt	L_∞ error
2	1	10^{-2}	5.0654×10^{-2}
	2	10^{-3}	4.8252×10^{-4}
3	2	10^{-2}	1.2459×10^{-5}
	3	10^{-3}	2.5194×10^{-7}

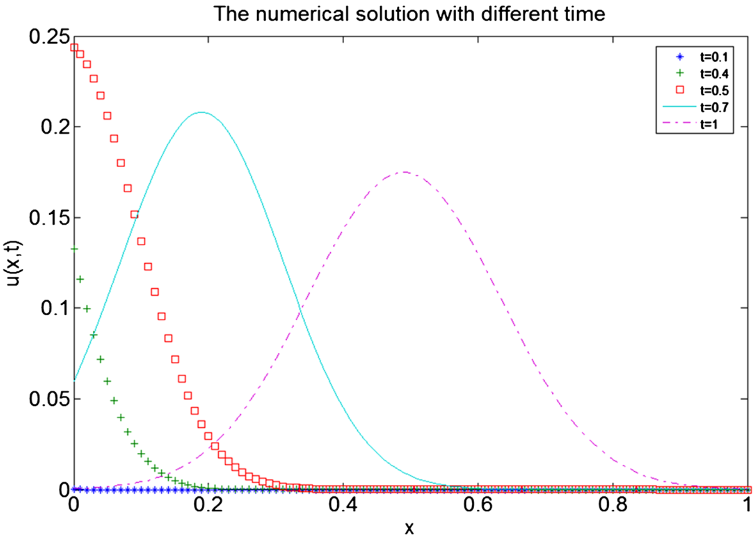

Figure 2. The numerical solution of Equation (6.1).

Acknowledgements

This work is supported by Technological Research Program of Chongqing Municipal Education Commission (Grant No. KJ130818 and KJ130810) and is funded by Fundamental and Advanced Research Project of Chongqing CSTC of China, the project No. are CSTC2013JCYJA00022 and cstc2012jjA00018.

References

[1] Murray, J.D. (2002) Mathematical Biology I: An Introduction. *Vol.* 17 *of Interdisciplinary Applied Mathematics*, 3rd Edition, Springer, New York.

[2] Wang, X.Y., Zhu, Z.S. and Lu, Y.K. (1990) Solitary Wave Solutions of the Generalised Burgers-Huxley Equation, *Journal of Physics A*, **23**, 271-274. http://dx.doi.org/10.1088/0305-4470/23/3/011

[3] Isenberg, J. and Gutfinger, C. (1973) Heat Transfer to a Draining Film. *International Journal of Heat and Mass Transfer*, **16**, 505-512. http://dx.doi.org/10.1016/0017-9310(73)90075-6

[4] Dehghan, M. (2004) Weighted Finite Difference Techniques for One Dimensional Advection-Difffusion Equation. *Applied Mathematics and Computation*, **147**, 307-319. http://dx.doi.org/10.1016/S0096-3003(02)00667-7

[5] Karahan, H. (2007) Unconditional Stable Explicit Finite Difference Technique for the Advection-Diffusion Equation Using Spreadsheets. *Advances in Engineering Software*, **38**, 80-84. http://dx.doi.org/10.1016/j.advengsoft.2006.08.001

[6] Liang, D. and Zhao, W. (1997) A High-Order Upwind Method for the Convection-Diffusion Problem. *Computer Methods in Applied Mechanics and Engineering*, **147**, 105-115. http://dx.doi.org/10.1016/S0045-7825(97)00004-2

[7] Dehghan, M. (2005) On the Numerical Solution of the One Dimensional Convection-Diffusion Equation. *Mathematical Problems in Engineering*, **1**, 61-74. http://dx.doi.org/10.1155/MPE.2005.61

[8] Golbabai, A. and Javidi, M. (2009) A Spectral Domain Decomposition Approach for the Generalized Burger's-Fisher Equation. *Chaos, Solitons & Fractals*, **39**, 385-392. http://dx.doi.org/10.1016/j.chaos.2007.04.013

[9] Mohammadi, A., Manteghian, M. and Mohammadi, A. (2011) Numerical Solution of the One-Dimensional Advection-Diffusion Equation Using Simultaneously Temporal and Spatial Weighted Parameters. *Australian Journal of Basic and Applied Sciences*, **5**, 1536-1543.

[10] Kumar, A., Jaiswal, D.K. and Kumar, N. (2010) Analytical Solutions to One-Dimensional Advection-Diffusion Equation with Variable Coefficients in Semi-Infinite Media. *Journal of Hydrology*, **380**, 330-337. http://dx.doi.org/10.1016/j.jhydrol.2009.11.008

[11] Liu, H. and Yan, J. (2009) The Direct Discontinuous Galerkin (DDG) Methods for Diffusion Problems. *SIAM Journal on Numerical Analysis*, **47**, 475-694. http://dx.doi.org/10.1137/080720255

[12] Kumar, A., Jaiswal, D.K. and Kumar, N. (2009) Analytical Solutions of One-Dimensional Advection-Diffusion Equation with Variable Coefficients in a Finite Domain. *Journal of Earth System Science*, **118**, 539-549. http://dx.doi.org/10.1007/s12040-009-0049-y

[13] Liu, H. and Yan, J. (2010) The Direct Discontinuous Galerkin (DDG) Method for Diffusion with Interface Corrections. *Communications in Computational Physics*, **8**, 541-564. http://dx.doi.org/10.4208/cicp.010909.011209a

[14] Zhang, R.-P. and Zhang, L.-W. (2012) Direct Discontinuous Galerkin Method for the Generalized Burgers Fisher Equation. *Chinese Physics B*, **9**, Article ID: 090206. http://dx.doi.org/10.1088/1674-1056/21/9/090206

[15] Dahmen, W., Kurdila, J. and Oswald, P. (1997) Multiscale Wavelet Methods for Partial Differential Equations. Academic Press, San Diego.

[16] Razzaghi, M. and Yousefi, S. (2001) The Legendre Wavelets Operational Matrix of Integration. *International Journal of Systems Science*, **32**, 495-502. http://dx.doi.org/10.1080/00207720120227

[17] Yousefi, S. (2010) Legendre Multiwavelet Galerkin Method for Solving the Hyperbolic Telegraph Equation. *Numerical Methods for Partial Differential Equations*, **26**, 539-543.

[18] Gantumur, T. and Stevenson, R.P. (2006) Computation of Differential Operators in Wavelet Coordinates. *Mathematics of Computation*, **75**, 697-710. http://dx.doi.org/10.1090/S0025-5718-05-01807-7

[19] Aziz, I., Siraj-ul-Islam and Sarler, B. (2013) Wavelets Collocation Method for Numerical Solution of Elliptic Problems. *Applied Mathematical Modelling*, **37**, 676-694. http://dx.doi.org/10.1016/j.apm.2012.02.046

[20] Liandrat, J., Perrier, V. and Tchmitchian, Ph. (1992) Numerical Resolution of Nonlinear Partial Differential Equations Using Wavelet Approach. In: Ruskai, M., Beylkin, G., Coifman, R., Daubechies, I., Mallat, S., Meyer, Y. and Raphael, L., Eds., *Wavelets and Their Applications*, Jones and Bartlett, Boston, 227-238.

[21] Kumar, R.B.V. and Mehra, M. (2006) A Three-Step Wavelet Galerkin Method for Parabolic and Hyperbolic Partial Differential Equations. *International Journal of Computer Mathematics*, **83**, 143-157. http://dx.doi.org/10.1080/00207160500112985

[22] Alpert, B., Beylkin, G., Gines, D. and Vozovoi, L. (2002) Adaptive Solution Partial Differential Equations in Multiwavelet Bases. *Journal of Computational Physics*, **182**, 149-190. http://dx.doi.org/10.1006/jcph.2002.7160

[23] Berlkin, G. (1992) On the Representation of Operators in Bases of Compactly Supported Wavelets. *SIAM Journal on Numerical Analysis*, **6**, 1716-1739. http://dx.doi.org/10.1137/0729097

[24] Zheng, X., Yang, X., Su, H. and Qiu, L. (2011) Discontinuous Legendre Wavelet Element Method for Elliptic Partial Differential Equations. *Applied Mathematics and Computation*, **218**, 3002-3018. http://dx.doi.org/10.1016/j.amc.2011.08.045

[25] Kima, M.-Y. and Wheelerb, M.F. (2014) A Multiscale Discontinuous Galerkin Method for Convection-Diffusion-Reaction Problems. *Computers and Mathematics with Applications*, **68**, 2251-2261. http://dx.doi.org/10.1016/j.camwa.2014.08.007