NASA/CR-2001-211026
ICASE Report No. 2001-20

A Local Discontinuous Galerkin Method for KdV-type Equations

Jue Yan and Chi-Wang Shu
Brown University, Providence, Rhode Island

ICASE
NASA Langley Research Center
Hampton, Virginia

Operated by Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NAS1-97046

June 2001

A LOCAL DISCONTINUOUS GALERKIN METHOD FOR KDV-TYPE EQUATIONS*

JUE YAN† AND CHI-WANG SHU‡

Abstract. In this paper we develop a local discontinuous Galerkin method for solving KdV type equations containing third derivative terms in one and two space dimensions. The method is based on the framework of the discontinuous Galerkin method for conservation laws and the local discontinuous Galerkin method for viscous equations containing second derivatives, however the guiding principle for inter-cell fluxes and nonlinear stability is new. We prove L^2 stability and a cell entropy inequality for the square entropy for a class of nonlinear PDEs of this type both in one and multiple spatial dimensions, and give an error estimate for the linear cases in the one dimensional case. The stability result holds in the limit case when the coefficients to the third derivative terms vanish, hence the method is especially suitable for problems which are "convection dominate", i.e. those with small second and third derivative terms. Numerical examples are shown to illustrate the capability of this method. The method has the usual advantage of local discontinuous Galerkin methods, namely it is extremely local and hence efficient for parallel implementations and easy for h-p adaptivity.

Key words. discontinuous Galerkin method, KdV equation, stability, error estimate

Subject classification. Applied and Numerical Mathematics

1. Introduction. In this paper we develop a local discontinuous Galerkin method for solving KdV type equations containing third derivative terms in one and multiple spatial dimensions. An example of such PDEs is the original KdV equation [18]

$$u_t + (\alpha u + \beta u^2)_x + \sigma u_{xxx} = 0, \tag{1.1}$$

where α, β and σ are constants. Our scheme can be designed and proven stable for more general nonlinearities, namely

$$u_t + f(u)_x + (r'(u)g(r(u)_x)_x)_x = 0 \tag{1.2}$$

in one space dimension for arbitrary (smooth) functions f, g and r, where $r'(u) = \frac{dr(u)}{du}$, and

$$u_t + \sum_{i=1}^{d} f_i(u)_{x_i} + \sum_{i=1}^{d} \left(r_i'(u) \sum_{j=1}^{d} g_{ij} (r_i(u)_{x_i})_{x_j} \right)_{x_i} = 0 \tag{1.3}$$

in multiple spatial dimensions for arbitrary (smooth) functions f_i, g_{ij} and r_i.

KdV type equations describe the propagation of waves in a variety of nonlinear, dispersive media and appear often in applications. See, e.g. [1]. Various numerical methods have been proposed and used in practice to solve this type of equations, see, e.g. [4, 5, 17]. However, in many situations, such as in the quantum hydrodynamic models of semiconductor device simulations [15] and in the dispersive limit of

*Research supported by ARO grant DAAD19-00-1-0405, NSF grants DMS-9804985 and ECS-9906606, NASA Langley grant NCC1-01035 and Contract NAS1-97046 while the second author was in residence at ICASE, NASA Langley Research Center, Hampton, VA 23681-2199, and AFOSR grant F49620-99-1-0077.

†Division of Applied Mathematics, Brown University, Providence, RI 02912. E-mail: yjue@cfm.brown.edu

‡Division of Applied Mathematics, Brown University, Providence, RI 02912. E-mail: shu@cfm.brown.edu

conservation laws [19], the third derivative terms might have small or even zero coefficients in some parts of the domain. We will call such cases as "convection dominated". The design of stable, efficient and high order methods, especially those for the "convection dominated" cases, i.e. when the third derivative terms are small ($|\sigma| \ll 1$ in (1.1)), remains a challenge.

The discontinuous Galerkin method is a class of finite element methods using completely discontinuous piecewise polynomial space for the numerical solution and the test functions. One certainly needs to use more degrees of freedom because of the discontinuities at the element boundaries, however this also gives one a room to design suitable inter-element boundary treatments (the so-called fluxes) to obtain highly accurate and stable methods in many difficult situations.

The first discontinuous Galerkin method was introduced in 1973 by Reed and Hill [20], in the framework of neutron transport (steady state linear hyperbolic equations). A major development of the discontinuous Galerkin method was carried out by Cockburn et al. in a series of papers [10, 9, 7, 11], in which they established a framework to easily solve *nonlinear* time dependent hyperbolic conservation laws (i.e. (1.2) and (1.3) without the third derivative terms) using explicit, nonlinearly stable high order Runge-Kutta time discretizations [22] and discontinuous Galerkin discretization in space with exact or approximate Riemann solvers as interface fluxes and TVB (total variation bounded) nonlinear limiters [21] to achieve non-oscillatory properties for strong shocks.

The discontinuous Galerkin method has found rapid applications in such diverse areas as aeroacoustics, electro-magnetism, gas dynamics, granular flows, magneto-hydrodynamics, meteorology, modeling of shallow water, oceanography, oil recovery simulation, semiconductor device simulation, transport of contaminant in porous media, turbomachinery, turbulent flows, viscoelastic flows and weather forecasting, among many others. Good references for the discontinuous Galerkin method and its recent development include the survey paper [8], other papers in that Springer volume, and the review paper [13].

The original discontinuous Galerkin method was designed to solve first order hyperbolic problems. A simple example to illustrate the essential ideas is the linear transport equation

$$u_t + u_x = 0. \tag{1.4}$$

Let's denote the mesh by $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, for $j = 1, ..., N$, with the center of the cell denoted by $x_j = \frac{1}{2}\left(x_{j-\frac{1}{2}} + x_{j+\frac{1}{2}}\right)$ and the size of each cell by $\Delta x_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$. We will denote $\Delta x = \max_j \Delta x_j$. If we multiply (1.4) by an arbitrary test function $v(x)$, integrate over the interval I_j, and integrate by parts, we get

$$\int_{I_j} u_t v dx - \int_{I_j} u v_x dx + u(x_{j+\frac{1}{2}}, t) v(x_{j+\frac{1}{2}}) - u(x_{j-\frac{1}{2}}, t) v(x_{j-\frac{1}{2}}) = 0. \tag{1.5}$$

This is the starting point for designing the discontinuous Galerkin method. We replace both the solution u and the test function v by piecewise polynomials of degree at most k. That is, $u, v \in V_{\Delta x}$ where

$$V_{\Delta x} = \{v : \ v \text{ is a polynomial of degree at most } k \text{ for } x \in I_j, \ \ j = 1, ..., N\}. \tag{1.6}$$

With this choice, there is an ambiguity in (1.5) in the last two terms involving the boundary values at $x_{j\pm\frac{1}{2}}$, as both the solution u and the test function v are *discontinuous* exactly at these boundary points. The idea is to treat these terms by an upwinding mechanism (information from characteristics), borrowed from successful high resolution finite volume schemes. Thus u at the interfaces $x_{j\pm\frac{1}{2}}$ is given by a single valued numerical flux $\hat{u}_{j\pm\frac{1}{2}} = u^-_{j\pm\frac{1}{2}}$, determined from upwinding, and v at the interfaces $x_{j\pm\frac{1}{2}}$ are given by the

values taken from inside the cell I_j, namely $v^-_{j+\frac{1}{2}}$ and $v^+_{j-\frac{1}{2}}$. Notice that we use v^- and v^+ to denote the left and right limits of v, respectively, at the interface where v is discontinuous. For more general nonlinear fluxes $f(u)$, the only difference is that the single valued flux $\hat{f}_{j+\frac{1}{2}}$ would be taken as a monotone flux depending on both $u^-_{j+\frac{1}{2}}$ and on $u^+_{j+\frac{1}{2}}$ (exact or approximate Riemann solvers in the system case). The resulting method of the lines ODE is then discretized by the nonlinearly stable high order Runge-Kutta time discretizations [22]. Nonlinear TVB limiters [21] may be used if the solution contains strong discontinuities. The schemes thus obtained, for solving hyperbolic conservation laws ((1.2) and (1.3) without the third derivative terms), have the following attractive properties:

1. It can be easily designed for any order of accuracy. In fact, the order of accuracy can be locally determined in each cell, thus allowing for efficient p adaptivity.
2. It can be used on arbitrary triangulations, even those with hanging nodes, thus allowing for efficient h adaptivity.
3. It is extremely local in data communications. The evolution of the solution in each cell needs to communicate only with the immediate neighbors, regardless of the order of accuracy, thus allowing for efficient parallel implementations. See, e.g. [3].
4. It has excellent provable nonlinear stability. One can prove a strong L^2 stability and a cell entropy inequality for the square entropy, for the general nonlinear cases, for any orders of accuracy on arbitrary triangulations in any space dimension, without the need for nonlinear limiters [16].

In [12] these discontinuous Galerkin methods were generalized to solve convection diffusion problems containing second derivative terms. This was motivated by the successful numerical experiments of Bassi and Rebay [2] for the compressible Navier-Stokes equations. The idea can be illustrated with the simple heat equation

$$u_t - u_{xx} = 0 \tag{1.7}$$

which we rewrite into a first order system

$$u_t - q_x = 0, \qquad q - u_x = 0, \tag{1.8}$$

we can then *formally* use the same discontinuous Galerkin method for the convection equation to solve (1.8), resulting in the following scheme: find $u, q \in V_{\Delta x}$ such that, for all test functions $v, w \in V_{\Delta x}$,

$$\begin{aligned} \int_{I_j} u_t v dx + \int_{I_j} q v_x dx - \hat{q}_{j+\frac{1}{2}} v^-_{j+\frac{1}{2}} + \hat{q}_{j-\frac{1}{2}} v^+_{j-\frac{1}{2}} = 0 \\ \int_{I_j} q w dx + \int_{I_j} u w_x dx - \hat{u}_{j+\frac{1}{2}} w^-_{j+\frac{1}{2}} + \hat{u}_{j-\frac{1}{2}} w^+_{j-\frac{1}{2}} = 0. \end{aligned} \tag{1.9}$$

However, there is no longer a upwinding mechanism or characteristics to guide the design of the fluxes $\hat{u}_{j+\frac{1}{2}}$ and $\hat{q}_{j+\frac{1}{2}}$. The crucial part in designing a stable and accurate algorithm (1.9) is a correct design of these fluxes. In [12], criteria are given for these fluxes to guarantee stability, convergence and a sub-optimal error estimate of order k for piecewise polynomials of degree k. The (most natural) central fluxes

$$\hat{u}_{j+\frac{1}{2}} = \frac{1}{2}\left(u^-_{j+\frac{1}{2}} + u^+_{j+\frac{1}{2}}\right), \qquad \hat{q}_{j+\frac{1}{2}} = \frac{1}{2}\left(q^-_{j+\frac{1}{2}} + q^+_{j+\frac{1}{2}}\right) \tag{1.10}$$

would satisfy these criteria and give a scheme which is indeed sub-optimal in the order of accuracy for odd k (i.e. the accuracy is order k rather than the expected order $k+1$ for odd k). This deficiency however is easily removed by going to a clever choice of fluxes, proposed in [12]:

$$\hat{u}_{j+\frac{1}{2}} = u^-_{j+\frac{1}{2}}, \qquad \hat{q}_{j+\frac{1}{2}} = q^+_{j+\frac{1}{2}}. \tag{1.11}$$

i.e. we alternatively take the left and right limits for the fluxes in u and q (we could of course also take the pair $u^+_{j+\frac{1}{2}}$ and $q^-_{j+\frac{1}{2}}$ as the fluxes). Notice that the evaluation of (1.11) is simpler than that of the central fluxes in (1.10), and this easily generalizes to multi space dimensions on arbitrary triangulations. The accuracy now becomes the optimal order $k+1$ for both even and odd k.

The schemes thus designed for the heat equation (1.7), or in fact for the most general multi dimensional nonlinear convection diffusion equations (nonlinear both in the first derivative convection part and the second derivation diffusion part), retain *all* the four nice properties listed above for the method used on convection equations. Moreover, the appearance of the auxiliary variable q is superficial: when a local basis is chosen in cell I_j then q is eliminated and the actual scheme for u takes a form similar to that for convection alone. This is a big advantage of the scheme over the traditional "mixed methods", and it is the reason that the scheme is termed *local* discontinuous Galerkin method in [12]. Even though the auxiliary variable q can be locally eliminated, it does approximate the derivative of the solution u to the same order of accuracy, thus matching the advantage of traditional "mixed methods" on this.

The purpose of this paper is to develop a similar local discontinuous Galerkin (LDG) method for the KdV like equations (1.1), (1.2) and (1.3) containing third derivative terms. Our objective is to design the method to retain again all the four nice properties listed above for the method used on convection and convection-diffusion equations, plus the feature that the method is local, namely the auxiliary variables introduced to approximate the first and second derivatives of the solution could be locally eliminated.

We will give a "preview" of the method on the simple linear equation

$$u_t + u_{xxx} = 0 \tag{1.12}$$

which we again rewrite into a first order system

$$u_t + p_x = 0, \qquad p - q_x = 0, \qquad q - u_x = 0. \tag{1.13}$$

We can then *formally* use the same discontinuous Galerkin method for the convection equation to solve (1.13), resulting in the following scheme: find $u, p, q \in V_{\Delta x}$ such that, for all test functions $v, w, z \in V_{\Delta x}$,

$$\begin{aligned}
&\int_{I_j} u_t v dx - \int_{I_j} p v_x dx + \hat{p}_{j+\frac{1}{2}} v^-_{j+\frac{1}{2}} - \hat{p}_{j-\frac{1}{2}} v^+_{j-\frac{1}{2}} = 0,\\
&\int_{I_j} p w dx + \int_{I_j} q w_x dx - \hat{q}_{j+\frac{1}{2}} w^-_{j+\frac{1}{2}} + \hat{q}_{j-\frac{1}{2}} w^+_{j-\frac{1}{2}} = 0,\\
&\int_{I_j} q z dx + \int_{I_j} u z_x dx - \hat{u}_{j+\frac{1}{2}} z^-_{j+\frac{1}{2}} + \hat{u}_{j-\frac{1}{2}} z^+_{j-\frac{1}{2}} = 0.
\end{aligned} \tag{1.14}$$

However, the fluxes $\hat{p}_{j+\frac{1}{2}}$, $\hat{q}_{j+\frac{1}{2}}$ and $\hat{u}_{j+\frac{1}{2}}$ must be designed based on different guiding principles than the first order convection or second order diffusion cases. The crucial part in designing a stable and accurate algorithm (1.14) is again a correct design of these fluxes. It turns out that the simple choices

$$\hat{p}_{j+\frac{1}{2}} = p^+_{j+\frac{1}{2}}, \qquad \hat{q}_{j+\frac{1}{2}} = q^+_{j+\frac{1}{2}}, \qquad \hat{u}_{j+\frac{1}{2}} = u^-_{j+\frac{1}{2}}, \tag{1.15}$$

would guarantee stability and convergence, as can be proven later in this paper and also clearly seen in Table 1.1, which contains numerical L^2 and L^∞ errors and orders of accuracy for the computed solution u for the method (1.14) with the fluxes (1.15) solving the equation (1.12) with an initial condition $u(x,0) = \sin(x)$ over the interval $[0, 2\pi]$ and periodic boundary conditions, at $t = 1$, using uniform meshes.

TABLE 1.1

$u_t + u_{xxx} = 0$. $u(x,0) = \sin(x)$. Periodic boundary conditions. L^2 and L^∞ errors. Uniform meshes with N cells. LDG methods with $k = 0, 1, 2, 3$. $t = 1$.

k		**N=10**	**N=20**		**N=40**		**N=80**	
		error	error	order	error	order	error	order
0	L^2	2.2534E-01	1.2042E-01	0.91	6.2185E-02	0.95	3.1582E-02	0.98
	L^∞	4.3137E-01	2.1977E-01	0.97	1.1082E-01	0.98	5.5376E-02	1.00
1	L^2	1.7150E-02	4.2865E-03	2.00	1.0716E-03	2.00	2.6792E-04	1.99
	L^∞	5.8467E-02	1.5757E-02	1.89	4.0487E-03	1.96	1.0210E-03	1.99
2	L^2	8.5803E-04	1.0823E-04	2.98	1.3559E-05	2.99	1.6958E-06	3.00
	L^∞	4.0673E-03	5.1029E-04	2.99	6.4490E-05	2.98	8.0722E-06	3.00
3	L^2	3.3463E-05	2.1035E-06	3.99	1.3166E-07	3.99	8.2365E-09	3.99
	L^∞	1.8185E-04	1.1157E-05	3.97	7.2362E-07	3.99	4.5593E-08	3.99

We remark that the choice for the fluxes (1.15) is not unique. In fact, the crucial part is to take $\hat{p}$ and $\hat{u}$ from opposite sides and to take $\hat{q}$ from the right. Thus

$$\hat{p}_{j+\frac{1}{2}} = p^-_{j+\frac{1}{2}}, \qquad \hat{q}_{j+\frac{1}{2}} = q^+_{j+\frac{1}{2}}, \qquad \hat{u}_{j+\frac{1}{2}} = u^+_{j+\frac{1}{2}},$$

would also work.

The organization of the paper is as follows. In section 2 we describe the method for the one dimensional case, and prove its nonlinear L^2 stability and a cell entropy inequality, as well as an error estimate for the linear case. In section 3 multiple spatial dimensional case is considered, where the nonlinear stability is given for the general triangulations. In section 4 we provide several numerical examples to illustrate the capability of the method. Concluding remarks and remarks about future work are given in section 5.

2. The LDG method for the one dimensional case. In this section, we present and analyze the LDG method for the following one dimensional nonlinear problem:

$$u_t + f(u)_x + (r'(u)g(r(u)_x)_x)_x = 0, \quad 0 \le x \le 1 \tag{2.1}$$

with an initial condition

$$u(x,0) = u^0(x), \quad 0 \le x \le 1 \tag{2.2}$$

and periodic boundary conditions. Here $f(u)$, $r(u)$ and $g(q)$ are arbitrary (smooth) nonlinear functions. Notice that the assumption of periodic boundary conditions is for simplicity only and is not essential: the method can be easily designed for non-periodic boundary conditions. Also notice that the linear equation (1.12) and the KdV equation (1.1) are both special cases of (2.1).

To define the LDG method, we first introduce the new variables

$$q = r(u)_x, \qquad p = g(q)_x \tag{2.3}$$

and rewrite the equation (2.1) as a first order system:

$$u_t + (f(u) + r'(u)p)_x = 0, \quad p - g(q)_x = 0, \qquad q - r(u)_x = 0. \tag{2.4}$$

The LDG method is obtained by discretizing the above system with the discontinuous Galerkin method. This is achieved by multiplying the three equations in (2.4) by three test functions v, w, z respectively, integrate over the interval I_j, and integrate by parts. We also need to pay special attention to the boundary terms resulting from the procedure of integration by parts, as mentioned in the previous section. Thus we seek piecewise polynomial solutions $u, p, q \in V_{\Delta x}$, where $V_{\Delta x}$ is defined in (1.6), such that for all test functions $v, w, z \in V_{\Delta x}$ we have, for $1 \le j \le N$,

$$\begin{aligned}
\int_{I_j} u_t v dx - \int_{I_j} (f(u) + r'(u)p) v_x dx + \left(\hat{f} + \widehat{r'}\hat{p}\right)_{j+\frac{1}{2}} v^-_{j+\frac{1}{2}} - \left(\hat{f} + \widehat{r'}\hat{p}\right)_{j-\frac{1}{2}} v^+_{j-\frac{1}{2}} = 0, \\
\int_{I_j} pw dx + \int_{I_j} g(q) w_x dx - \hat{g}_{j+\frac{1}{2}} w^-_{j+\frac{1}{2}} + \hat{g}_{j-\frac{1}{2}} w^+_{j-\frac{1}{2}} = 0, \\
\int_{I_j} qz dx + \int_{I_j} r(u) z_x dx - \hat{r}_{j+\frac{1}{2}} z^-_{j+\frac{1}{2}} + \hat{r}_{j-\frac{1}{2}} z^+_{j-\frac{1}{2}} = 0.
\end{aligned} \tag{2.5}$$

Notice that we still use letters without a subscript Δx to denote functions in the finite element space $V_{\Delta x}$, to simplify the notations. The only ambiguity in the algorithm (2.5) now is the definition of the numerical fluxes (the "hats"), which should be designed based on different guiding principles than the first order convection or second order diffusion cases to ensure stability. It turns out that we can take the simple choices (we omit the subscripts $j \pm \frac{1}{2}$ in the definition of the fluxes as all quantities are evaluated at the interfaces $x_{j\pm\frac{1}{2}}$)

$$\hat{f} = \hat{f}(u^-, u^+), \quad \widehat{r'} = \frac{r(u^+) - r(u^-)}{u^+ - u^-}, \quad \hat{p} = p^+, \quad \hat{g} = \hat{g}(q^-, q^+), \quad \hat{r} = r(u^-) \tag{2.6}$$

where $\hat{f}(u^-, u^+)$ is a monotone flux for $f(u)$, namely $\hat{f}(u^-, u^+)$ is a Lipschitz continuous function in both arguments u^- and u^+, is consistent with $f(u)$ in the sense that $\hat{f}(u, u) = f(u)$, and is a non-decreasing function in u^- and a non-increasing function in u^+. Likewise, $-\hat{g}(q^-, q^+)$ is a monotone flux for $-g(q)$, namely $\hat{g}(q^-, q^+)$ is a Lipschitz continuous function in both arguments q^- and q^+, is consistent with $g(q)$ in the sense that $\hat{g}(q, q) = g(q)$, and is a non-increasing function in q^- and a non-decreasing function in q^+. Examples of monotone fluxes which are suitable for discontinuous Galerkin methods can be found in, e.g. [10]. We could for example use the simple Lax-Friedrichs flux

$$\hat{f}(u^-, u^+) = \frac{1}{2}\left(f(u^-) + f(u^+) - \alpha(u^+ - u^-)\right), \quad \alpha = \max_u |f'(u)|. \tag{2.7}$$

where the maximum is taken over a relevant range of u. The algorithm is now well defined.

We remark that the choice for the fluxes (2.6) is not unique. In fact, the crucial part is to take $\hat{p}$ and $\hat{r}$ from opposite sides. Thus

$$\hat{f} = \hat{f}(u^-, u^+), \quad \widehat{r'} = \frac{r(u^+) - r(u^-)}{u^+ - u^-}, \quad \hat{p} = p^-, \quad \hat{g} = \hat{g}(q^-, q^+), \quad \hat{r} = r(u^+)$$

would also work.

We also remark that the algorithm (2.5)-(2.6) is very easy for numerical implementation. Given u, one first uses the third equation in (2.5) to obtain q. This is achieved locally: q in I_j can be obtained with the information of u in the cells I_j and I_{j-1}. The second equation in (2.5) is then used to obtain p locally: p in I_j can be obtained with the information of q in (at most) the cells I_j, I_{j-1} and I_{j+1}. Finally, the update of the solution u is obtained using the first equation in (2.5), again locally, namely the update of u in I_j can be obtained with the information of u in (at most) the cells I_j, I_{j-1} and I_{j+1} and that of p in the cells I_j and I_{j+1}.

We have the following "cell entropy inequality" for the scheme (2.5)-(2.6). This is a generalization of the cell entropy inequality obtained in [16] for the discontinuous Galerkin method applied to hyperbolic conservation laws (equation (2.1) with $g(q) = 0$).

Proposition 2.1. (cell entropy inequality) There exist numerical entropy fluxes $\hat{H}_{j+\frac{1}{2}}$ such that the solution to the scheme (2.5)-(2.6) satisfies

$$\frac{d}{dt}\int_{I_j}\left(\frac{u^2(x,t)}{2}\right)dx + \left(\hat{H}_{j+\frac{1}{2}} - \hat{H}_{j-\frac{1}{2}}\right) \leq 0. \tag{2.8}$$

Proof: We sum up the three equalities in (2.5) and introduce the notation

$$\begin{aligned} B_j(u,p,q;v,w,z) = &\int_{I_j} u_t v dx - \int_{I_j}(f(u) + r'(u)p)v_x dx + \left(\hat{f} + \widehat{r'}\hat{p}\right)_{j+\frac{1}{2}} v^-_{j+\frac{1}{2}} \\ &- \left(\hat{f} + \widehat{r'}\hat{p}\right)_{j-\frac{1}{2}} v^+_{j-\frac{1}{2}} + \int_{I_j} pw dx + \int_{I_j} g(q)w_x dx - \hat{g}_{j+\frac{1}{2}} w^-_{j+\frac{1}{2}} \\ &+\hat{g}_{j-\frac{1}{2}} w^+_{j-\frac{1}{2}} + \int_{I_j} qz dx + \int_{I_j} r(u)z_x dx - \hat{r}_{j+\frac{1}{2}} z^-_{j+\frac{1}{2}} + \hat{r}_{j-\frac{1}{2}} z^+_{j-\frac{1}{2}}. \end{aligned} \tag{2.9}$$

Clearly, the solutions u, p, q of the scheme (2.5)-(2.6) satisfy

$$B_j(u,p,q;v,w,z) = 0 \tag{2.10}$$

for all $v, w, z \in V_{\Delta x}$. We then take

$$v = u, \qquad w = q, \qquad z = -p$$

to obtain, after some algebraic manipulations,

$$0 = B_j(u,p,q;u,q,-p) = \frac{d}{dt}\int_{I_j}\left(\frac{u^2(x,t)}{2}\right)dx + \left(\hat{H}_{j+\frac{1}{2}} - \hat{H}_{j-\frac{1}{2}}\right) + \Theta_{j-\frac{1}{2}}$$

with the numerical entropy flux $\hat{H}$ defined by

$$\hat{H} = -F(u^-) + G(q^-) - r(u^-)p^- + \left(\hat{f} + \widehat{r'}\hat{p}\right)u^- - \hat{g}q^- + \hat{r}p^-$$

and the extra term Θ given by

$$\Theta = [F(u) - G(q) + r(u)p] - \left(\hat{f} + \widehat{r'}\hat{p}\right)[u] + \hat{g}[q] - \hat{r}[p],$$

Here

$$F(u) = \int^u f(u)du, \qquad G(q) = \int^q g(q)dq,$$

and

$$[v] = v^+ - v^-$$

denotes the jump of v. Notice that we have dropped the subscripts about the location $j - \frac{1}{2}$ or $j + \frac{1}{2}$ as all these quantities are defined at a single interface and depend only on the left and right values at that interface. Now all we need to do is to verify $\Theta \geq 0$. To this end, we notice that, with the definition (2.6) of the numerical fluxes and with simple algebraic manipulations, we easily obtain

$$[r(u)p] - \widehat{r'}\hat{p}[u] - \hat{r}[p] = 0$$

and hence

$$\begin{aligned}\Theta &= [F(u)] - \hat{f}[u] - [G(q)] + \hat{g}[q] \\ &= \int_{u^-}^{u^+} \left(f(s) - \hat{f}(u^-, u^+) \right) ds - \int_{q^-}^{q^+} \left(g(s) - \hat{g}(q^-, q^+) \right) ds \\ &\geq 0,\end{aligned} \tag{2.11}$$

where the last inequality follows from the monotonicity of the fluxes $\hat{f}$ and $-\hat{g}$. This finishes the proof. □

Now the L^2 stability of the method is a trivial corollary:

Corollary 2.1. (L^2 stability) The solution to the scheme (2.5)-(2.6) satisfies the L^2 stability

$$\frac{d}{dt} \int_0^1 \left(\frac{u^2(x,t)}{2} \right) dx \leq 0. \tag{2.12}$$

Proof: We simply add up (2.8) over j. □

About time discretizations, if we denote the semi-discrete LDG method (2.5)-(2.6) by

$$u_t = R(u),$$

then the following implicit θ scheme

$$\frac{u^{n+1} - u^n}{\Delta t} = R(u^{n+\theta}), \qquad u^{n+\theta} = (1-\theta)u^n + \theta u^{n+1} \tag{2.13}$$

will also satisfy the same cell entropy inequality and L^2 stability as long as $\frac{1}{2} \leq \theta \leq 1$. Notice that this includes the first order backward Euler and second order Crank-Nicolson implicit time discretizations as special cases. See [16] for the purely hyperbolic case.

Proposition 2.2. (implicit time discretization) The cell entropy inequality and the L^2 stability also hold for the time discretization (2.13) with $\frac{1}{2} \leq \theta \leq 1$ for the scheme (2.5)-(2.6). That is,

$$\int_{I_j} \left(\frac{(u^{n+1}(x))^2 - (u^n(x))^2}{2\Delta t} \right) dx + \hat{H}^{n+\theta}_{j+\frac{1}{2}} - \hat{H}^{n+\theta}_{j-\frac{1}{2}} \leq 0, \tag{2.14}$$

and

$$\int_0^1 (u^{n+1}(x))^2 dx \leq \int_0^1 (u^n(x))^2 dx. \tag{2.15}$$

Proof: If we take the test functions at $n+\theta$, e.g. $v = u^{n+\theta}$ given by (2.13), we obtain just as before

$$\int_{I_j} \frac{u^{n+1}(x) - u^n(x)}{\Delta t} u^{n+\theta} dx + \hat{H}^{n+\theta}_{j+\frac{1}{2}} - \hat{H}^{n+\theta}_{j-\frac{1}{2}} \leq 0,$$

which can be rewritten as

$$\int_{I_j} \left(\frac{(u^{n+1}(x))^2 - (u^n(x))^2}{2\Delta t} \right) dx + \hat{H}^{n+\theta}_{j+\frac{1}{2}} - \hat{H}^{n+\theta}_{j-\frac{1}{2}} + \left(\theta - \frac{1}{2} \right) \int_{I_j} \left(\frac{(u^{n+1}(x) - u^n(x))^2}{\Delta t} \right) dx \leq 0.$$

Thus, a sufficient condition to get the cell entropy inequality (2.14) is just $\theta \geq \frac{1}{2}$. Again, (2.15) is obtained simply by adding up (2.14) over j. □

The stability result obtained here can be used to get an error estimate in L^2 for the numerical solution u, when the equation (2.1) is linear. Without loss of generality we may take $f(u) = u$, $g(q) = q$ and $r(u) = u$, resulting in the equation

$$u_t^e + u_x^e + u_{xxx}^e = 0. \tag{2.16}$$

Noticed that we have used the notation u^e to denote the exact solution of the PDE in order not to confuse with the numerical solutions. We have the following result, where C here and below denotes a generic constant which may be of different values at different locations.

Proposition 2.3. (error estimate) The error for the scheme (2.5)-(2.6) applied to the linear PDE (2.16) satisfies

$$\sqrt{\int_0^1 \left(u^e(x,t) - u(x,t)\right)^2 dx} \le C\Delta x^{k+\frac{1}{2}}, \tag{2.17}$$

where the constant C depends on the derivatives of u^e and time t.

Proof: First, we notice that, in this linear case, most monotone fluxes simply become upwinding

$$\hat{f}(u^-, u^+) = u^-, \qquad \hat{g}(q^-, q^+) = q^+,$$

and this is what we will assume. It is then easy to work out the exact form of Θ in (2.11) for the cell entropy inequality to be

$$\Theta = \frac{1}{2}\left([u]^2 + [q]^2\right). \tag{2.18}$$

We now notice that the exact solution of the PDE (2.16), u^e, $q^e = u_x^e$ and $p^e = u_{xx}^e$ clearly satisfies

$$B_j(u^e, p^e, q^e; v, w, z) = 0$$

for all $v, w, z \in V_{\Delta x}$, where B_j is defined by (2.9). Taking the difference between the above equality and (2.10), we obtain the *error equation*

$$B_j(u^e - u, p^e - p, q^e - q; v, w, z) = 0 \tag{2.19}$$

for all $v, w, z \in V_{\Delta x}$. As usual this error equation is the basic starting point of error estimates.

We now take

$$v = \mathcal{S}u^e - u, \qquad w = \mathcal{P}q^e - q, \qquad z = p - \mathcal{P}p^e, \tag{2.20}$$

in the error equation (2.19). Here $\mathcal{P}$ is the standard L^2 projection into $V_{\Delta x}$, that is, for each j,

$$\int_{I_j} (\mathcal{P}w(x) - w(x))v(x)dx = 0 \qquad \forall v \in P^k,$$

where P^k denotes the space of all polynomials of degree at most k. In other words, the difference between the projection $\mathcal{P}w$ and the original function w is orthogonal to all polynomials of degree up to k in each interval. $\mathcal{S}$ is a special projection into $V_{\Delta x}$ which satisfies, for each j,

$$\int_{I_j} (\mathcal{S}w(x) - w(x))v(x)dx = 0 \quad \forall v \in P^{k-1} \quad \textit{and} \quad \mathcal{S}w(x_{j+1/2}^-) = w(x_{j+1/2}^-),$$

in other words, the difference between the projection $\mathcal{S}w$ and the original function w is orthogonal to all polynomials of degree up to $k-1$ in each interval, and the projection agrees with the function at the right boundary in each interval. This special projection is needed for u because we have no control on the jumps of p in the cell entropy inequality, see (2.18). Substituting (2.20) into the error equation (2.19) and moving terms, we obtain

$$B_j(v,-z,w;v,w,z) = B_j(v^e,-z^e,w^e;v,w,z) \tag{2.21}$$

where v, w, z are given by (2.20), and v^e, w^e, z^e are given by

$$v^e = \mathcal{S}u^e - u^e, \qquad w^e = \mathcal{P}q^e - q^e, \qquad z^e = p^e - \mathcal{P}p^e. \tag{2.22}$$

By the same argument as that used for the cell entropy inequality, the left hand side of (2.21) becomes

$$B_j(v,-z,w;v,w,z) = \frac{d}{dt}\int_{I_j}\left(\frac{v^2}{2}\right)dx + \left(\hat{H}_{j+\frac{1}{2}} - \hat{H}_{j-\frac{1}{2}}\right) + \Theta_{j-\frac{1}{2}} \tag{2.23}$$

where, by (2.18),

$$\Theta_{j-\frac{1}{2}} = \frac{1}{2}\left([v]^2_{j-\frac{1}{2}} + [w]^2_{j-\frac{1}{2}}\right). \tag{2.24}$$

The right hand side of (2.21) can be written out as

$$B_j(v^e,-z^e,w^e;v,w,z) = \mathcal{I} + \mathcal{II} + \mathcal{III} + \mathcal{IV} \tag{2.25}$$

where

$$\mathcal{I} = \int_{I_j} v^e_t v dx, \tag{2.26}$$

$$\mathcal{II} = -\int_{I_j} z^e w dx + \int_{I_j} w^e z dx - \int_{I_j}(v^e - z^e)v_x dx + \int_{I_j} w^e w_x dx + \int_{I_j} v^e z_x dx, \tag{2.27}$$

$$\mathcal{III} = -\left(\left(v^e_{j-\frac{1}{2}}\right)^- - \left(z^e_{j-\frac{1}{2}}\right)^+\right)[v]_{j-\frac{1}{2}} + \left(w^e_{j-\frac{1}{2}}\right)^+ [w]_{j-\frac{1}{2}} + \left(v^e_{j-\frac{1}{2}}\right)^- [z]_{j-\frac{1}{2}}, \tag{2.28}$$

and

$$\mathcal{IV} = \hat{h}_{j+\frac{1}{2}} - \hat{h}_{j-\frac{1}{2}} \tag{2.29}$$

for some flux function $\hat{h}$. Notice that v, w, z are given by (2.20) and v^e, w^e, x^e are given by (2.22), respectively.

Now, by using the simple inequality $ab \le \frac{1}{2}(a^2+b^2)$, and standard approximation theory on $v^e_t = (\mathcal{S}u^e - u^e)_t$, see, e.g. [6], we have

$$\mathcal{I} \le C\Delta x_j^{2k+3} + \int_{I_j}\left(\frac{v^2}{2}\right)dx.$$

Because $\mathcal{P}$ is a local L^2 projection, and $\mathcal{S}$, even though not a local L^2 projection, does have the property that $w - \mathcal{S}w$ is locally orthogonal to all polynomials of degree up to $k-1$, all the terms in $\mathcal{II}$ are actually zero. The last term in $\mathcal{III}$ is zero, because of the special interpolating property of the projection $\mathcal{S}$. An application

of the simple inequality $ab \le \frac{1}{2}(a^2+b^2)$ for the first two terms in $\mathcal{III}$ and standard approximation theory on the point values of $v^e - z^e = (\mathcal{S}u^e - u^e) + (\mathcal{P}p^e - p^e)$ and of $w^e = \mathcal{P}q^e - q^e$ (see, e.g. [6]) then gives

$$\mathcal{III} \le C(\Delta x_{j-1}^{2k+2} + \Delta x_j^{2k+2} + \frac{1}{4}\left([v]^2 + [w]^2\right).$$

Finally, $\mathcal{IV}$ only contains flux difference terms which will vanish upon a summation in j.

Combining all these and summing over j we obtain the following inequality

$$\frac{d}{dt}\int_0^1 \left(\frac{v^2}{2}\right) dx + \frac{1}{4}\left([v]^2 + [w]^2\right) \le C\Delta x^{2k+1} + \int_0^1 \left(\frac{v^2}{2}\right) dx.$$

An integration in t plus the standard approximation theory on $v^e = \mathcal{S}u^e - u^e$ then gives the desired error estimate (2.17). □

3. The LDG method for the multiple dimensional case. In this section, we generalize the scheme discussed in the previous section to multiple spatial dimensions $x = (x_1, \cdots, x_d)$. We solve the following nonlinear problem:

$$u_t + \sum_{i=1}^d f_i(u)_{x_i} + \sum_{i=1}^d \left(r_i'(u) \sum_{j=1}^d g_{ij}(r_i(u)_{x_i})_{x_j} \right)_{x_i} = 0, \quad 0 \le x_i \le 1, \quad i = 1, \cdots, d \tag{3.1}$$

with an initial condition

$$u(x,0) = u^0(x), \quad 0 \le x_i \le 1, \quad i = 1, \cdots, d \tag{3.2}$$

and periodic boundary conditions. Here $f_i(u)$, $r_i(u)$ and $g_{ij}(q)$ are arbitrary (smooth) nonlinear functions. Notice that the assumption of a box geometry and periodic boundary conditions is for simplicity only and is not essential: the method can be easily designed for arbitrary domain and for non-periodic boundary conditions.

Let's denote a triangulation of the unit box by $\mathcal{T}_{\Delta x}$, consisting of non-overlapping polyhedra covering completely the unit box. Hanging nodes are allowed. Here Δx measures the longest edge of all polyhedra in $\mathcal{T}_{\Delta x}$. We again denote the finite element space by

$$V_{\Delta x}^d = \{v : \ v \text{ is a polynomial of degree at most } k \text{ for } x \in K, \ \forall K \in \mathcal{T}_{\Delta x}\}. \tag{3.3}$$

Similar to the one dimensional case, to define the LDG method, we first introduce the new variables

$$q_i = r_i(u)_{x_i}, \qquad p_i = \sum_{j=1}^d g_{ij}(q_i)_{x_j}, \qquad i = 1, \cdots, d \tag{3.4}$$

and rewrite the equation (3.1) as a first order system:

$$\begin{aligned} &u_t + \sum_{i=1}^d (f_i(u) + r_i'(u)p_i)_{x_i} = 0, \\ &p_i - \sum_{j=1}^d g_{ij}(q_i)_{x_j} = 0, \qquad q_i - r_i(u)_{x_i} = 0, \qquad i = 1, \cdots, d. \end{aligned} \tag{3.5}$$

The LDG method is obtained by discretizing the above system with the discontinuous Galerkin method. This is achieved by multiplying the equations in (3.5) by test functions v, w_i, z_i respectively, integrate over

an element $K \in \mathcal{T}_{\Delta x}$, and integrate by parts. We again need to pay special attention to the boundary terms resulting from the procedure of integration by parts, as in the one dimensional case. Thus we seek piecewise polynomial solutions $u, p_i, q_i \in V^d_{\Delta x}$, where $V^d_{\Delta x}$ is defined in (3.3), such that for all test functions $v, w_i, z_i \in V^d_{\Delta x}$ we have

$$\begin{aligned}
&\int_K u_t v dx - \sum_{i=1}^d \int_K (f_i(u) + r_i'(u)p_i) v_{x_i} dx + \int_{\partial K} \widehat{h_{n_K}}\, v^{int_K} ds = 0, \\
&\int_K p_i w_i dx + \sum_{j=1}^d \int_K g_{ij}(q_i)(w_i)_{x_j} dx - \int_{\partial K} \widehat{g_{i,n_K}}\, w^{int_K} ds = 0, \qquad i = 1, \cdots, d \\
&\int_K q_i z_i dx + \int_K r_i(u)(z_i)_{x_i} dx - \int_{\partial K} \widehat{r_{i,n_K}}\, z^{int_K} ds = 0, \qquad i = 1, \cdots, d,
\end{aligned} \tag{3.6}$$

where ∂K is the boundary of element K, and the numerical fluxes (the "hats") are defined similar to the one dimensional cases, namely

$$\begin{aligned}
\widehat{h_{n_K}} &= \widehat{f_{n_K,K}}(u^{int_K}, u^{ext_K}) + \frac{\sum_{i=1}^d \left(r_i(u^{ext_K}) - r_i(u^{int_K})\right)\, p_i^+\, n_{i,K}}{u^{ext_K} - u^{int_K}} \\
\widehat{g_{i,n_K}} &= \widehat{g_{i,n_K,K}}(q^{int_K}, q^{ext_K}), \qquad \widehat{r_{i,n_K}} = r_i(u^-)\, n_{i,K}.
\end{aligned} \tag{3.7}$$

Here $n_K = (n_{1,K}, \cdots, n_{d,K})$ is the outward unit normal for element K along the element boundary ∂K, u^{int_K} denotes the value of u evaluated from inside the element K, and u^{ext_K} denotes the value of u evaluated from outside the element K (inside the neighboring element). On the other hand, p^+ denotes the value of p evaluated from a pre-designated "plus" side along an edge e, which is always the boundaries of two neighboring elements. For example, we could choose a fixed vector β which is not parallel with any normals of element boundaries, and then designate the "plus" side to be the side at the end of the arrow of the normal n with $n \cdot \beta > 0$, see Figure 3.1. $\widehat{f_{n_K,K}}(u^{int_K}, u^{ext_K})$ is a monotone flux for $f_{n_K}(u) = \sum_{i=1}^d f_i(u) n_{i,K}$, namely $\widehat{f_{n_K,K}}(u^{int_K}, u^{ext_K})$ is a Lipschitz continuous function in both arguments u^{int_K} and u^{ext_K}, is consistent with $f_{n_K}(u)$ in the sense that $\widehat{f_{n_K}}(u,u) = f_{n_K}(u)$, and is a non-decreasing function in u^{int_K} and a non-increasing function in u^{ext_K}. Moreover, it is conservative (that is, there is only one flux at each edge shared by two elements, added to the residue for one and subtracted from the reside for another), namely

$$\widehat{f_{n_K,K}}(a,b) = -\widehat{f_{n_{K'},K'}}(b,a)$$

where K and K' share the same edge where the flux is computed and hence $n_{K'} = -n_K$. Likewise, $-\widehat{g_{i,n_K,K}}(q_i^{int_K}, q_i^{ext_K})$ is a monotone flux for $-g_{i,n_K}(q_i) = -\sum_{j=1}^d g_{ij}(q) n_{j,K}$. Notice that we can again use the one dimensional monotone fluxes as in the previous section. For example, we can use the simple Lax-Friedrichs flux

$$\begin{aligned}
&\widehat{f_{n_K,K}}(u^{int_K}, u^{ext_K}) = \frac{1}{2}\left(\sum_{i=1}^d \left(f_i(u^{int_K}) + f_i(u^{ext_K})\right)\, n_{i,K} - \alpha(u^{ext_K} - u^{int_K})\right), \\
&\alpha = \max_u |f_{n_K}'(u)|,
\end{aligned} \tag{3.8}$$

where the maximum is taken over a relevant range of u. The algorithm is now well defined.

Again, the algorithm (3.6)-(3.7) is very easy for numerical implementation. Given u, one first locally solves for the q_i, then locally solves for the p_i, and finally locally solves for the update of u. All the advantages listed for the method for the one dimensional case are still valid in this multiple dimensional case.

We still have the following "cell entropy inequality" for the scheme (3.6)-(3.7). The proof follows the same lines as that for the one dimensional case, so we will omit it.

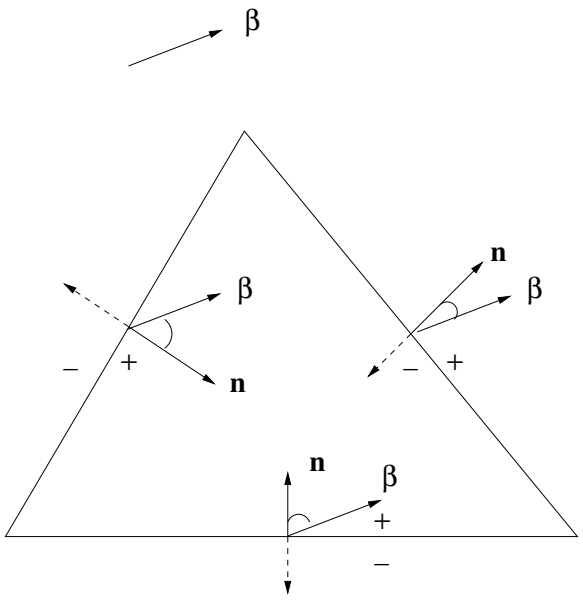

FIG. 3.1. *Illustration of the definition of "plus" and "minus" sides determined by a pre-determined vector* β.

Proposition 3.1. (cell entropy inequality) There exist conservative numerical entropy fluxes $\widehat{H_{n_K,K}}$ such that the solution to the scheme (3.6)-(3.7) satisfies

$$\frac{d}{dt}\int_K \left(\frac{u^2(x,t)}{2}\right) dx + \int_{\partial K} \widehat{H_{n_K,K}} ds \leq 0. \tag{3.9}$$

□

The L^2 stability of the method is then again a trivial corollary, by summing up the cell entropy inequalities over K:

Corollary 3.1. (L^2 stability) The solution to the scheme (3.6)-(3.7) satisfies the L^2 stability

$$\frac{d}{dt}\int_\Omega \left(\frac{u^2(x,t)}{2}\right) dx \leq 0. \tag{3.10}$$

□

The same cell entropy inequality also holds for the implicit time discretizations:

Proposition 3.2. (implicit time discretization) The cell entropy inequality and the L^2 stability also hold for the time discretization (2.13) with $\frac{1}{2} \leq \theta \leq 1$ for the scheme (3.6)-(3.7). That is,

$$\int_K \left(\frac{(u^{n+1}(x))^2 - (u^n(x))^2}{2\Delta t}\right) dx + \int_{\partial K} \widehat{H^{n+\theta}_{n_K,K}} ds \leq 0, \tag{3.11}$$

and

$$\int_\Omega (u^{n+1}(x))^2 dx \leq \int_\Omega (u^n(x))^2 dx. \tag{3.12}$$

□

Unfortunately, we could not get the optimal error estimate because of the lack of a suitable projection $\mathcal{S}$ similar to the one dimensional case. However, numerical examples in the next section verify that the accuracy holds as in the one dimensional case.

TABLE 4.1

$u_t + u_{xxx} = 0$. $u(x,0) = \sin(x)$. Periodic boundary conditions. L^2 and L^∞ errors. Non-uniform meshes with N cells. LDG methods with $k = 0, 1, 2, 3$. $t = 1$.

k		N=10	N=20		N=40		N=80	
		error	error	order	error	order	error	order
0	L^2	2.2222E-01	1.2014E-01	0.88	6.2532E-02	0.94	3.1900E-02	0.97
	L^∞	4.3282E-01	2.2006E-01	0.97	1.1210E-01	0.97	5.8810E-02	0.93
1	L^2	2.0144E-02	5.2347E-03	1.94	1.3322E-03	1.97	3.3592E-04	1.98
	L^∞	8.8110E-02	2.3302E-02	1.93	5.9387E-03	1.97	1.4969E-03	1.98
2	L^2	9.8394E-04	1.1974E-04	3.03	1.4953E-05	3.00	1.8687E-06	3.00
	L^∞	5.2984E-03	6.8421E-04	2.95	8.5138E-05	3.00	1.0728E-05	2.99
3	L^2	7.3589E-05	4.6509E-06	3.98	2.9191E-06	3.99	2.0141E-08	3.86
	L^∞	3.4438E-04	2.2260E-05	3.95	1.3992E-06	3.99	9.1039E-08	3.94

4. Numerical examples. In this section we provide a few preliminary numerical examples to illustrate the accuracy and capability of the method. Attention has not been paid to efficiency in time discretizations, so explicit third order Runge-Kutta method [22] is used. Study of suitable implicit time discretizations which have efficient iterative solvers maintaining the local structure of the method is the subject of a future study.

We would like to illustrate through these numerical examples the high order accuracy of the methods for both one dimensional and two dimensional, both linear and nonlinear problems. We would also like to illustrate the good behavior of the method for the so-called convection dominated cases, namely the case where the coefficients of the third derivative terms are small.

Example 4.1. We compute the solution of the linear one dimensional equation

$$u_t + u_{xxx} = 0 \tag{4.1}$$

with an initial condition $u(x,0) = \sin(x)$ and periodic boundary conditions (with 2π periodicity). The exact solution is given by $u(x,t) = \sin(x+t)$. Both uniform meshes and non-uniform meshes are used. The non-uniform meshes in this and later examples are a repeated pattern of $0.9\Delta x$ and $1.1\Delta x$ with an even number of elements. The L^2 and L^∞ errors and the numerical order of accuracy are contained in Table 1.1 (in section 1) for the uniform mesh case, and in Table 4.1 for the non-uniform mesh case. We can clearly see that the method with P^k elements are giving a uniform $(k+1)$-th order of accuracy in both norms for both the uniform and the non-uniform meshes.

Example 4.2. We compute the solution of the linear two dimensional equation

$$u_t + u_{xxx} + u_{yyy} = 0 \tag{4.2}$$

with an initial condition $u(x,y,0) = \sin(x+y)$ and periodic boundary conditions (with 2π periodicity) in both directions. The exact solution is given by $u(x,y,t) = \sin(x+y+2t)$. Both uniform and non-uniform rectangular meshes are used. The non-uniform meshes are a repeated pattern of $0.9\Delta x$ and $1.1\Delta x$, in both directions, with an even number of edges in both directions. The L^2 and L^∞ errors and the numerical order of accuracy are contained in Table 4.2 for the uniform mesh case, and in Table 4.3 for the non-uniform mesh

TABLE 4.2

$u_t + u_{xxx} + u_{yyy} = 0$. $u(x, y, 0) = \sin(x + y)$. Periodic boundary conditions. L^2 and L^∞ errors. Uniform meshes with $N \times N$ cells. LDG methods with $k = 0, 1, 2, 3$. $t = 1$.

k		**10×10**	**20×20**		**40×40**	
		error	error	order	error	order
0	L^2	3.5528E-01	2.0535E-01	0.79	1.1090E-01	0.89
	L^∞	7.1359E-01	4.0190E-01	0.82	2.1165E-01	0.92
1	L^2	3.3603E-02	9.0904E-03	1.89	2.4084E-03	1.92
	L^∞	2.2074E-01	6.1899E-02	1.83	1.5962E-02	1.95
2	L^2	3.8750E-03	4.8463E-04	2.99	6.0501E-05	3.00
	L^∞	3.9084E-02	4.8902E-03	2.99	6.1274E-04	2.99
3	L^2	4.1491E-04	2.6426E-05	3.97	1.6550E-06	3.99
	L^∞	4.2847E-03	2.8478E-04	3.91	1.7846E-05	3.99

TABLE 4.3

$u_t + u_{xxx} + u_{yyy} = 0$. $u(x, y, 0) = \sin(x + y)$. Periodic boundary conditions. L^2 and L^∞ errors. Non-uniform meshes with $N \times N$ cells. LDG methods with $k = 0, 1, 2, 3$. $t = 1$.

k		**10×10**	**20×20**		**40×40**	
		error	error	order	error	order
0	L^2	3.5963E-01	2.0788E-01	0.79	1.1228E-01	0.88
	L^∞	7.3869E-01	4.0713E-01	0.85	2.1681E-01	0.91
1	L^2	3.4590E-02	9.1681E-03	1.92	2.3412E-03	1.97
	L^∞	2.5815E-01	7.2978E-02	1.82	1.8533E-02	1.97
2	L^2	4.0949E-03	5.1285E-04	2.99	6.4054E-05	3.00
	L^∞	5.0429E-02	6.3078E-03	2.99	8.0584E-04	2.97
3	L^2	4.5434E-04	2.8854E-05	3.97	1.8080E-06	3.99
	L^∞	6.0982E-03	4.0321E-04	3.92	2.5340E-05	3.99

case. We can clearly see again that the method with P^k elements are giving a uniform $(k+1)$-th order of accuracy for both the uniform and the non-uniform meshes.

Example 4.3. In order to see the accuracy of the method for nonlinear problems, we compute the classical soliton solution of the KdV equation

$$u_t - 3\left(u^2\right)_x + u_{xxx} = 0 \tag{4.3}$$

in $-10 \leq x \leq 12$. The initial condition is given by

$$u(x, 0) = -2\,\mathrm{sech}^2(x),$$

The exact solution is

$$u(x, t) = -2\,\mathrm{sech}^2(x - 4t).$$

TABLE 4.4

The KdV equation $u_t - 3\left(u^2\right)_x + u_{xxx} = 0$. $u(x,0) = -2\,sech^2(x)$. *Boundary condition (4.4).* L^2 *and* L^∞ *errors. Uniform meshes with* N *cells. LDG methods with* $k = 0, 1, 2, 3$. $t = 0.5$.

k		**N=40**	**N=80**		**N=160**		**N=320**	
		error	error	order	error	order	error	order
0	L^2	2.5292E-01	1.9098E-01	0.40	1.3019E-01	0.55	7.9780E-02	0.71
	L^∞	9.0170E-01	6.8651E-01	0.39	4.6405E-01	0.56	2.8531E-01	0.70
1	L^2	2.6512E-02	4.6652E-03	2.50	1.0108E-03	2.20	2.5906E-04	1.96
	L^∞	1.4748E-01	3.4625E-02	2.09	1.1840E-02	1.55	3.3239E-03	1.83
2	L^2	1.5317E-03	1.8083E-04	3.08	2.2642E-05	2.99	2.8335E-06	2.99
	L^∞	1.7486E-02	2.7505E-03	2.66	3.5575E-04	2.95	4.4397E-05	3.00
3	L^2	2.0631E-04	1.3981E-05	3.88	8.9054E-07	3.97	5.6029E-08	3.99
	L^∞	2.0155E-03	2.1462E-04	3.23	1.4461E-05	3.89	9.1140E-07	3.98

TABLE 4.5

The KdV equation $u_t - 3\left(u^2\right)_x + u_{xxx} = 0$. $u(x,0) = -2\,sech^2(x)$. *Boundary condition (4.4).* L^2 *and* L^∞ *errors. Non-uniform meshes with* N *cells. LDG methods with* $k = 0, 1, 2, 3$. $t = 0.5$.

k		**N=40**	**N=80**		**N=160**		**N=320**	
		error	error	order	error	order	error	order
0	L^2	2.4530E-01	1.9004E-01	0.37	1.3390E-01	0.50	8.4635E-02	0.66
	L^∞	1.0172E+00	7.6826E-01	0.40	5.3383E-01	0.52	3.3655E-01	0.66
1	L^2	2.7042E-02	4.9065E-03	2.46	1.0555E-03	2.21	2.6978E-04	1.97
	L^∞	1.4490E-01	4.1570E-02	1.80	1.3925E-02	1.57	3.9129E-03	1.83
2	L^2	1.9493E-03	2.0134E-04	3.27	2.4926E-05	3.01	3.1208E-06	2.99
	L^∞	2.2876E-02	3.5163E-03	2.70	4.7161E-04	2.89	5.9033E-05	2.99
3	L^2	3.0402E-04	1.5462E-05	4.29	1.0064E-06	3.94	6.3370E-08	3.99
	L^∞	2.7735E-03	2.1464E-04	3.69	1.8358E-05	3.55	1.3119E-06	3.80

We compute the solution with two different boundary conditions. Table 4.4 (uniform mesh) and Table 4.5 (non-uniform mesh) give the errors of numerical solution at $t = 0.5$ using the boundary condition

$$u(-10,t) = g_1(t), \qquad u_x(12,t) = g_2(t), \qquad u_{xx}(12,t) = g_3(t) \tag{4.4}$$

where $g_i(t)$ corresponds to the data from the exact solution. Notice that the LDG method allows an easy implementation of such boundary conditions. Table 4.6 (uniform mesh) and Table 4.7 (non-uniform mesh) give the errors of numerical solution using the periodic boundary conditions. Although the exact solution is not periodic, the large size of the computational domain allows the usage of periodic boundary conditions with negligible error. We can see from these tables that the orders of accuracy are comparable to that for the linear case.

TABLE 4.6

The KdV equation $u_t - 3\left(u^2\right)_x + u_{xxx} = 0$. $u(x,0) = -2\,sech^2(x)$. *Periodic boundary condition.* L^2 *and* L^∞ *errors. Uniform meshes with* N *cells. LDG methods with* $k = 0, 1, 2, 3$. $t = 0.5$.

k		N=40	N=80		N=160		N=320	
		error	error	order	error	order	error	order
0	L^2	2.5292E-01	1.9098E-01	0.40	1.3020E-01	0.55	7.9822E-02	0.70
	L^∞	9.0170E-01	6.8648E-01	0.39	4.6404E-01	0.56	2.8602E-01	0.69
1	L^2	2.6600E-02	4.6801E-03	2.50	1.0133E-03	2.20	2.5966E-04	1.96
	L^∞	1.4778E-01	3.4403E-02	2.10	1.1930E-02	1.52	3.3404E-03	1.84
2	L^2	1.5883E-03	1.8254E-04	3.12	2.2699E-05	3.00	2.8353E-06	3.00
	L^∞	1.7729E-02	2.7130E-03	2.70	3.5359E-04	2.94	4.4350E-05	2.99
3	L^2	2.1442E-04	1.5566E-05	3.78	1.0318E-06	3.91	6.5818E-08	3.97
	L^∞	1.9911E-03	2.2607E-04	3.14	1.5397E-05	3.88	9.7191E-07	3.98

TABLE 4.7

The KdV equation $u_t - 3\left(u^2\right)_x + u_{xxx} = 0$. $u(x,0) = -2\,sech^2(x)$. *Periodic boundary condition.* L^2 *and* L^∞ *errors. Non-uniform meshes with* N *cells. LDG methods with* $k = 0, 1, 2, 3$. $t = 0.5$.

k		N=40	N=80		N=160		N=320	
		error	error	order	error	order	error	order
0	L^2	2.4530E-01	1.9004E-01	0.37	1.3391E-01	0.50	8.4650E-02	0.66
	L^∞	1.0172E+00	7.6826E-01	0.40	5.3383E-01	0.52	3.3672E-01	0.66
1	L^2	2.7071E-02	4.9216E-03	2.46	1.0581E-03	2.21	2.7039E-04	1.97
	L^∞	1.4507E-01	4.1341E-02	1.81	1.3916E-02	1.57	3.9383E-03	1.82
2	L^2	2.0350E-03	2.0344E-04	3.32	2.4988E-05	3.02	3.1228E-06	3.00
	L^∞	2.2916E-02	3.4702E-03	2.72	4.6922E-04	2.88	5.8972E-05	2.99
3	L^2	3.2212E-04	1.8451E-05	4.12	1.1715E-06	3.97	7.4102E-08	3.98
	L^∞	2.8274E-03	2.2498E-04	3.65	1.9437E-05	3.53	1.3793E-06	3.81

Example 4.4. In order to see the accuracy of the method for nonlinear problems with small coefficient for the third derivative term, we compute the soliton solution of the generalized KdV equation [5]

$$u_t + u_x + \left(\frac{u^4}{4}\right)_x + \epsilon u_{xxx} = 0, \tag{4.5}$$

in $-2 \le x \le 3$, where we take $\epsilon = 0.2058 \times 10^{-4}$. The initial condition is given by

$$u(x,0) = A\,\mathrm{sech}^{\frac{2}{3}}\left(K(x - x_0)\right) \tag{4.6}$$

with $A = 0.2275$, $x_0 = 0.5$, and $K = 3\left(\frac{A^3}{40\epsilon}\right)^{\frac{1}{2}}$. The exact solution is

$$u(x,t) = A\,\mathrm{sech}^{\frac{2}{3}}\left(K(x - x_0) - \omega t\right)$$

where $\omega = K\left(1 + \frac{A^3}{10}\right)$. We compute the solution using the boundary condition

$$u(-2,t) = g_1(t), \qquad u_x(3,t) = g_2(t), \qquad u_{xx}(3,t) = g_3(t) \tag{4.7}$$

TABLE 4.8

The GKdV equation (4.5) with initial condition (4.6) and boundary condition (4.7). L^2 and L^∞ errors. Non-uniform meshes with N cells. LDG methods with $k = 0, 1, 2, 3$. $t = 1$.

k		**N=160**	**N=320**		**N=640**		**N=1280**	
		error	error	order	error	order	error	order
0	L^2	1.6566E-02	1.1259E-02	0.56	7.0817E-03	0.67	4.1526E-03	0.77
	L^∞	9.3056E-02	6.6829E-02	0.48	4.4502E-02	0.58	2.7539E-02	0.69
1	L^2	3.8554E-04	6.0675E-05	2.66	1.1784E-05	2.36	2.8635E-06	2.04
	L^∞	3.2635E-03	6.2508E-04	2.38	2.2689E-04	1.47	6.4595E-05	1.81
2	L^2	8.2907E-06	9.5348E-07	3.12	1.1895E-07	3.00	1.5290E-08	2.96
	L^∞	1.6684E-04	2.2545E-05	2.88	3.0858E-06	2.87	3.9503E-07	2.97
3	L^2	1.7005E-06	1.3664E-07	3.63	3.0527E-09	5.48	1.9206E-10	3.99
	L^∞	1.7607E-05	1.3291E-06	3.72	8.3962E-08	3.98	5.2861E-09	3.99

with a non-uniform mesh. The result is contained in Table 4.8.

Example 4.5. In this example we compute the classical soliton solutions of the KdV equation

$$u_t + \left(\frac{u^2}{2}\right)_x + \epsilon u_{xxx} = 0. \tag{4.8}$$

The examples are those used in [14].

The single soliton case has the initial condition

$$u_0(x) = 3c\,\mathrm{sech}^2\left(k(x - x_0)\right) \tag{4.9}$$

with $c = 0.3$, $x_0 = 0.5$, $k = (1/2)\sqrt{c/\epsilon}$ and $\epsilon = 5 \times 10^{-4}$. The solution is computed in $x \in [0, 2]$ with periodic boundary conditions, using P^2 elements with 100 cells, and is shown in Figure 4.1.

The double soliton collision case has the initial condition

$$u_0(x) = 3c_1\,\mathrm{sech}^2\left(k_1(x - x_1)\right) + 3c_2\,\mathrm{sech}^2\left(k_2(x - x_2)\right) \tag{4.10}$$

with $c_1 = 0.3$, $c_2 = 0.1$, $x_1 = 0.4$, $x_2 = 0.8$, $k_i = (1/2)\sqrt{c_i/\epsilon}$ for $i = 1, 2$, and $\epsilon = 4.84 \times 10^{-4}$. The solution is computed in $x \in [0, 2]$ with periodic boundary conditions, using P^2 elements with 100 cells. and is shown in Figure 4.2.

The triple soliton splitting case has the initial condition

$$u_0(x) = \frac{2}{3}\,\mathrm{sech}^2\left(\frac{x - 1}{\sqrt{108\epsilon}}\right) \tag{4.11}$$

with $\epsilon = 10^{-4}$. The solution is computed in $x \in [0, 3]$ with periodic boundary conditions and is shown in Figure 4.3.

Example 4.6. We compute in this example the KdV zero dispersion limit of conservation laws. The equation is (4.8) with an initial condition

$$u(x, 0) = 2 + 0.5\sin(2\pi x) \tag{4.12}$$

for $x \in [0, 1]$ with periodic boundary conditions, and we are interested in the limit when $\epsilon \to 0^+$. Theoretical and numerical discussions about this limit can be found in [19] and [23]. Here we are mainly concerned with

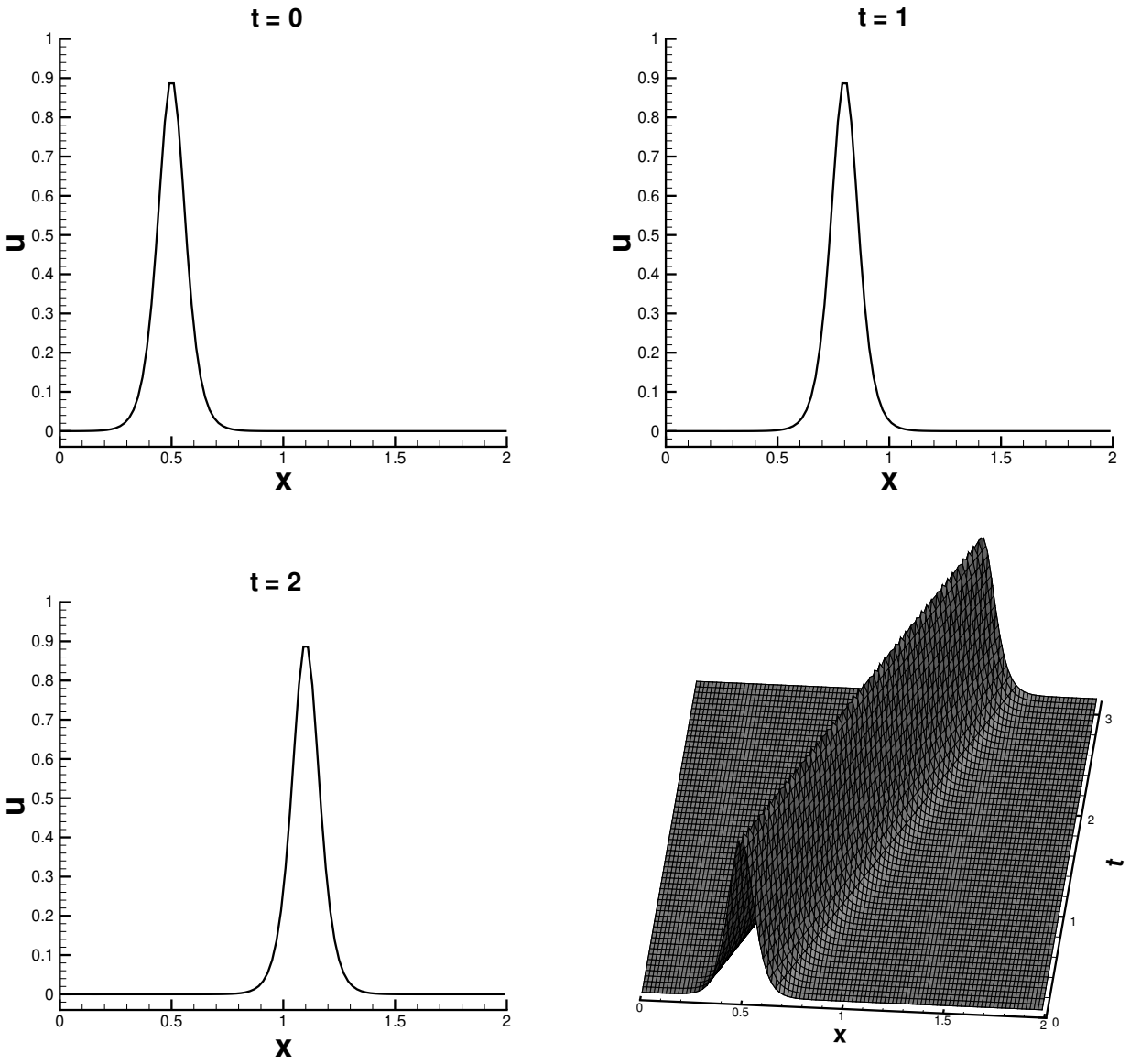

FIG. 4.1. *Single soliton profiles. Solutions of equation (4.8) with initial condition (4.9) and periodic boundary conditions in [0,2] using P^2 elements with 100 cells. Top left: solution at $t = 0$; top right: $t = 1$; bottom left: $t = 2$; bottom right: space time graph of the solution up to $t = 3$.*

the capability of our numerical method in resolving the small scale solution structures in this limit when ϵ is small. For this purpose we compute the solution to $t = 0.5$ with $\epsilon = 10^{-4}, 10^{-5}, 10^{-6}$ and 10^{-7} using P^2 elements with 300 cells for the first two cases, 800 cells for the third case and 1700 cells for the last case. We have verified that these are "converged" solutions in the sense that further increasing the number of cells does not change the solutions graphically. These solutions are shown in Figure 4.4. Notice the physical "oscillations" which are typical in such dispersive limits, see, e.g. [19]. Clearly our method is very suitable to compute such solutions.

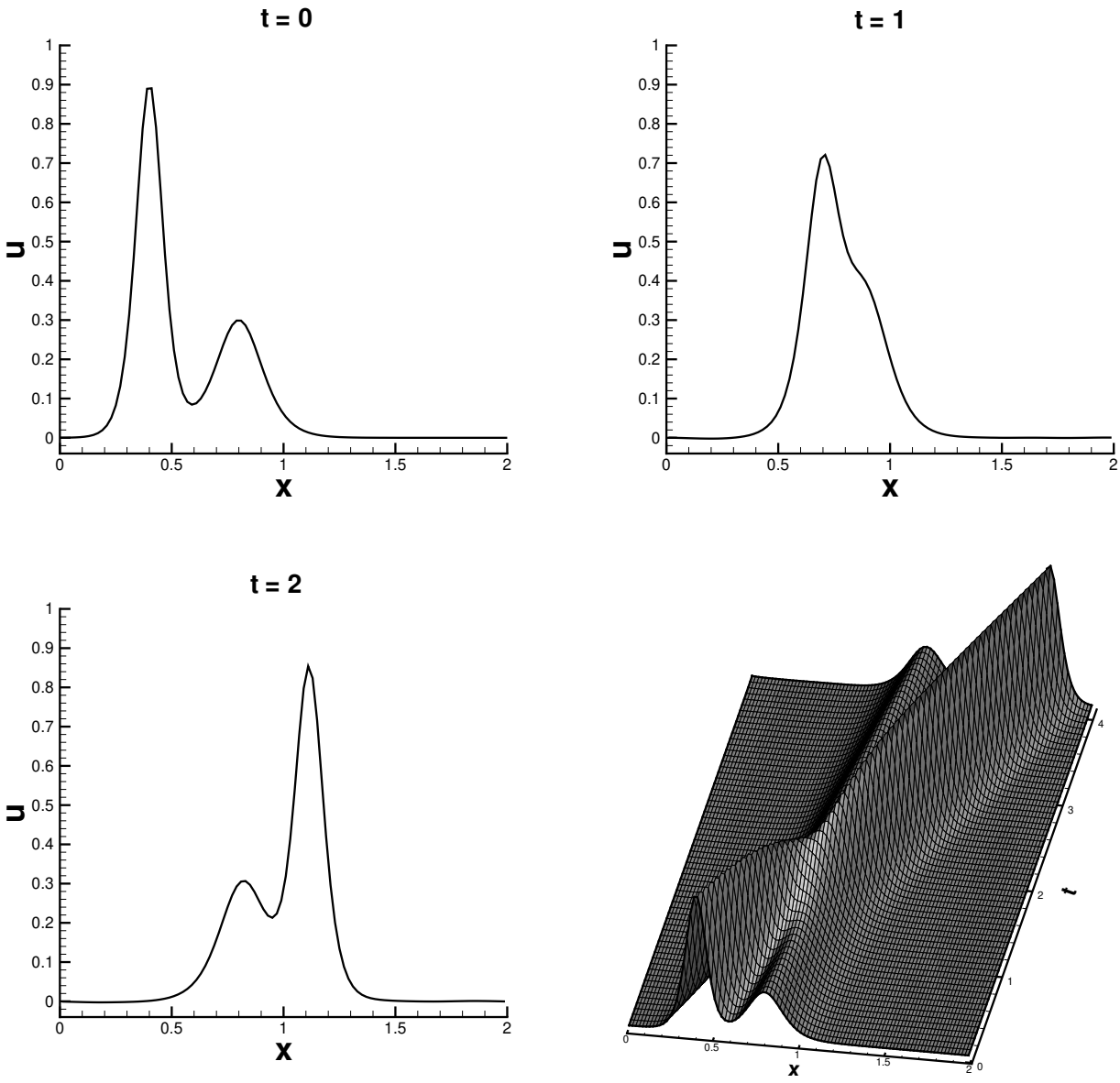

FIG. 4.2. *Double soliton collision profiles. Solutions of equation (4.8) with initial condition (4.10) and periodic boundary conditions in [0,2] using P^2 elements with 100 cells. Top left: solution at $t = 0$; top right: $t = 1$; bottom left: $t = 2$; bottom right: space time graph of the solution up to $t = 4$.*

5. Concluding remarks. We have designed a class of local discontinuous Galerkin methods for solving KdV type equations containing third derivatives and have proven their stability for any spatial dimensions for a general class of nonlinear equations. Numerical examples are shown to illustrate the accuracy and capability of the methods, especially for the convection dominated cases where the coefficients of the third derivative terms are small. Efficient implicit time discretizations which have efficient iterative solvers maintaining the local structure of the method, accuracy enhancement study, and more numerical experiments with physically interesting problems constitute an ongoing project.

Acknowledgments. We would like to thank Bernardo Cockburn for his valuable help in the discussion

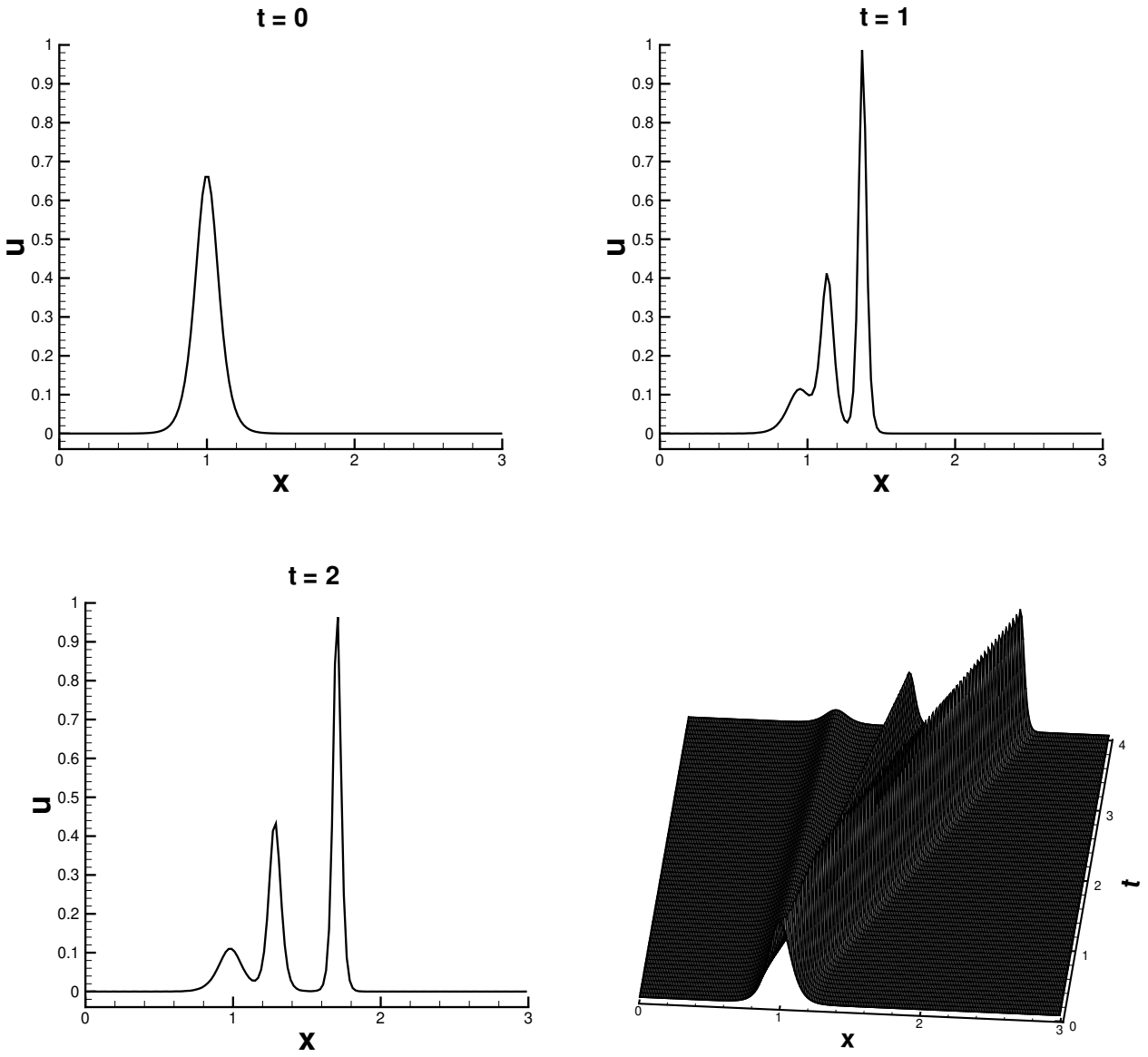

FIG. 4.3. *Triple soliton splitting profiles. Solutions of equation (4.8) with initial condition (4.11) and periodic boundary conditions in [0,3] using P^2 elements with 150 cells. Top left: solution at $t = 0$; top right: $t = 1$; bottom left: $t = 2$; bottom right: space time graph of the solution up to $t = 4$.*

about the projection $\mathcal{S}$, and Andy Majda for pointing out reference [19] and test cases of zero dispersive limits of conservation laws in Example 4.6.

REFERENCES

[1] T. B. BENJAMIN, J. L. BONA AND J. J. MAHONY, *Model equations for long waves in nonlinear, dispersive systems*, Phil. Trans. Roy. Soc. Lond. A, 272 (1972), pp. 47-78.

[2] F. BASSI AND S. REBAY, *A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations*, J. Comput. Phys., 131 (1997), pp. 267-279.

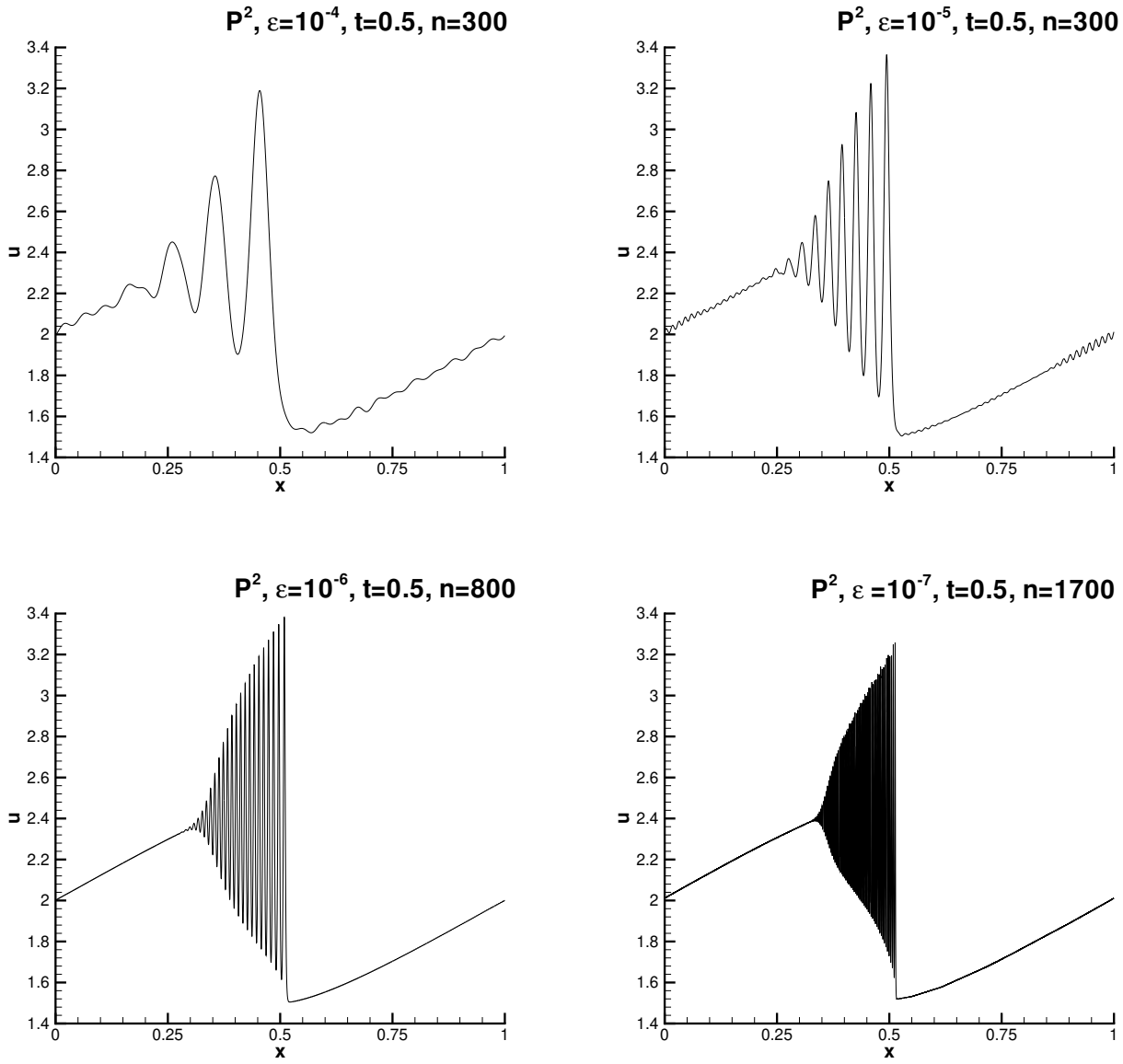

FIG. 4.4. *Zero dispersion limit of conservation laws. Solutions of equation (4.8) with initial condition (4.12) and periodic boundary conditions in [0,1] using P^2 elements at $t = 0.5$. Top left: $\epsilon = 10^{-4}$ with 300 cells; top right: $\epsilon = 10^{-5}$ with 300 cells; bottom left: $\epsilon = 10^{-6}$ with 800 cells; bottom right: $\epsilon = 10^{-7}$ with 1700 cells.*

[3] R. BISWAS, K. D. DEVINE AND J. FLAHERTY, *Parallel, adaptive finite element methods for conservation laws*, Appl. Numer. Math., 14 (1994), pp. 255-283.

[4] J. L. BONA, V. A. DOUGALIS AND O. A. KARAKASHIAN, *Fully discrete Galerkin methods for the Korteweg-de Vries equation*, Comput. Math. Appl., 12A (1986), pp. 859-884.

[5] J. L. BONA, V. A. DOUGALIS, O. A. KARAKASHIAN AND W. R. MCKINNEY, *Conservative, high-order numerical schemes for the generalized Korteweg-de Vries equation*, Phil. Trans. Roy. Soc. Lond. A, 351 (1995), pp. 107-164.

[6] P. CIARLET, *The finite element method for elliptic problems*, North Holland, 1975.

[7] B. COCKBURN, S. HOU, AND C.-W. SHU, *TVB Runge-Kutta local projection discontinuous Galerkin*

finite element method for conservation laws IV: the multidimensional case, Math. Comp., 54 (1990), pp. 545-581.

[8] B. Cockburn, G. Karniadakis and C.-W. Shu, *The development of discontinuous Galerkin methods*, in *Discontinuous Galerkin Methods: Theory, Computation and Applications*, B. Cockburn, G. Karniadakis and C.-W. Shu, editors, Lecture Notes in Computational Science and Engineering, volume 11, Springer, 2000, Part I: Overview, pp. 3-50.

[9] B. Cockburn, S.-Y. Lin and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems*, J. Comput. Phys., 84 (1989), pp. 90-113.

[10] B. Cockburn and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: general framework*, Math. Comp., 52 (1989), pp. 411-435.

[11] B. Cockburn and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws V: multidimensional systems*, J. Comput. Phys., 141 (1998), pp. 199-224.

[12] B. Cockburn and C.-W. Shu, *The local discontinuous Galerkin method for time-dependent convection diffusion systems*, SIAM J. Numer. Anal., 35 (1998), pp. 2440-2463.

[13] B. Cockburn and C.-W. Shu, *Runge-Kutta Discontinuous Galerkin methods for convection-dominated problems*, submitted to J. Sci. Comput.

[14] A. Debussche and J. Printems, *Numerical simulation of the stochastic Korteweg-de Vries equation*, Physica D, 134 (1999), pp. 200-226.

[15] C.L. Gardner, *The quantum hydrodynamic model for semiconductor devices*, SIAM J. Appl. Math., 54 (1994), pp. 409-427.

[16] G.-S. Jiang and C.-W. Shu, *On cell entropy inequality for discontinuous Galerkin methods*, Math. Comp., 62 (1994), pp. 531-538.

[17] O. A. Karakashian and W. R. McKinney, *On the approximation of solutions of the generalized Korteweg-de Vries equation*, Math. Comput. Simulation, 37 (1994), pp. 405-416.

[18] D. J. Korteweg and G. de Vries, *On the change of form of long waves advancing in a rectangular canal and on a new type of long stationary waves*, Philosophical Magazine, 39 (1895), pp. 422-443.

[19] P. D. Lax, C. D. Levermore and S. Venakides, *The generation and propagation of oscillations in dispersive initial value problems and their limiting behavior*, in *Important Developments in Soliton Theory*, Springer Series in Nonlinear Dynamics, A.S. Fokas and V.E. Zakharov, Eds., Springer-Verlag, 1993, pp. 205-241.

[20] W. H. Reed and T. R. Hill, *Triangular mesh methods for the neutron transport equation*, Tech. Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[21] C.-W. Shu, *TVB uniformly high-order schemes for conservation laws*, Math. Comp., 49 (1987), pp. 105-121.

[22] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock capturing schemes*, Journal of Computational Physics, 77 (1988), pp. 439-471.

[23] S. Venakides, *The zero dispersion limit of the Korteweg-de Vries equation with periodic initial data*, AMS Trans. 301 (1987), pp. 189-225.