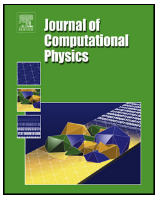
Journal of Computational Physics 313 (2016) 754–774

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp



A hybrid LDG-HWENO scheme for KdV-type equations ☆

Dongmi Luo [a], Weizhang Huang [b], Jianxian Qiu [a,*]

[a] School of Mathematical Sciences and Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computing, Xiamen University, Xiamen, Fujian 361005, China
[b] Department of Mathematics, University of Kansas, Lawrence, KS 66045, USA

ARTICLE INFO

Article history:
Received 13 November 2015
Received in revised form 27 February 2016
Accepted 27 February 2016
Available online 2 March 2016

Keywords:
Local discontinuous Galerkin method
Third order equations
High order method
HWENO
KdV equation

ABSTRACT

A hybrid LDG-HWENO scheme is proposed for the numerical solution of KdV-type partial differential equations. It evolves the cell averages of the physical solution and its moments (a feature of Hermite WENO) while discretizes high order spatial derivatives using the local DG method. The new scheme has the advantages of both LDG and HWENO methods, including the ability to deal with high order spatial derivatives and the use of a small number of global unknown variables. The latter is independent of the order of the scheme and the spatial order of the underlying differential equations. One and two dimensional numerical examples are presented to show that the scheme can attain the same formal high order accuracy as the LDG method.

1. Introduction

We consider the numerical solution of KdV-type equations in one and two spatial dimensions. The local discontinuous Galerkin (LDG) method for this type of partial differential equations (PDEs) has been studied in [21,23]. The LDG method is an extension of the discontinuous Galerkin (DG) method aimed at solving PDEs containing higher than first order spatial derivatives. The DG method was first introduced by Reed and Hill [16] for solving linear hyperbolic problems for neutron transfer. A major development of the DG method was carried out by Cockburn et al. in a series of papers [2–6].

The basic idea of the LDG method is to rewrite a PDE with high order spatial derivatives into a system of first order PDEs and then discretize it by the DG method. It can achieve nonlinear stability without slope limiters when carefully designed. The sub-optimal error estimates in L^2 norm were obtained for the smooth solution of linear equations and the cell entropy inequality was proven in [23]. While the LDG method also has many other advantages, it has the disadvantage of employing a large number of unknown variables and the number increases rapidly as the order of the method increases especially in multiple dimensions and for high order PDEs. On the other hand, the essentially non-oscillatory (ENO) schemes of Shu and Osher [19,20], the weighted ENO (WENO) schemes of Liu et al. [12] and Jiang and Shu [11], and Hermite WENO (HWENO) schemes of Qiu and Shu [14,15] have the advantage of employing a small number of unknown variables. But they have the disadvantage that the stencil used in reconstruction is becoming wider with an increasing order of accuracy.

☆ The research was supported by the NSFC grants 11328104, 91530107 and 11571290.

* Corresponding author.

E-mail addresses: luodongmi@126.com (D. Luo), whuang@ku.edu (W. Huang), jxqiu@xmu.edu.cn (J. Qiu).

http://dx.doi.org/10.1016/j.jcp.2016.02.064
0021-9991/© 2016 Elsevier Inc. All rights reserved.

To avoid the disadvantages of DG/LDG and WENO schemes, based on the reconstruction procedure for HWENO limiter by Qiu and Shu [14,15] a hybrid RKDG-HWENO scheme was proposed by Balsara et al. [1] for hyperbolic conservation laws. It has intrinsic robustness and smaller stencils. The scheme evolves lower moments while reconstructing the higher moments of the solution. Dumbser et al. [8] extended the scheme to a new family of in-cell recovery DG method, referred to as $P_N P_M$ methods, where P_N indicates that a piecewise polynomial of degree N is used to represent a DG solution, and P_M stands for a reconstructed polynomial solution of degree M ($M \geq N$) that is used to compute the numerical fluxes. This approach yields a general, unified framework that contains two important special cases, classical high order finite volume (FV) schemes ($N = 0$) and the conventional discontinuous Galerkin (DG) method ($N = M$), and has been applied successfully to simulate the Euler equations of compressible gas dynamics and the equations of ideal magnetohydrodynamics (MHD). $P_N P_M$ methods were extended to the numerical solution of the compressible Navier–Stokes equations [7,13].

Motivated by the hybrid RKDG-HWENO and $P_N P_M$ methods, we propose a hybrid LDG-HWENO scheme for the numerical solution of KdV-type PDEs. To our best knowledge, those methods have not been studied for higher order PDEs like KdV-type equations containing third order spatial derivatives. The new scheme employs LDG to approximate higher than first order spatial derivatives contained in the PDEs. Also different from HWENO methods in [14,15], where the cell averages of both the solution u and its first derivative u_x are evolved in time and used in the reconstruction, we use the cell averages of u (the solution) and $u\frac{x-x_j}{\Delta x}$ (the first moment(s)) which have been used in the reconstruction procedure for HWENO limiter [14,15]. Compared with LDG methods, the new scheme employs only a small number of (global) unknown variables and has the same stencils in the reconstruction. Moreover, since the solutions of KdV-type equations are smooth in general, only linear weights are used in our HWENO reconstruction (see Section 2), that is, the linear HWENO reconstruction is used in the new scheme. This makes the scheme more efficient because the computation of nonlinear weights typically takes a significant percentage of the total CPU time. Numerical examples show that the new scheme has a good long-term stability and can attain the same formal high order accuracy as the LDG method.

An outline of the paper is given as follows. The hybrid LDG-HWENO scheme is described in Sections 2 and 3 for one and two dimensional KdV-type equations, respectively. In Section 4, a selection of one and two dimensional numerical examples are presented to demonstrate the accuracy and the capability of the scheme. Conclusions are drawn in Section 5.

2. The hybrid LDG-HWENO scheme in one dimension

We consider the numerical solution of one dimensional KdV-type problems in the general form,

$$u_t + f(u)_x + (r'(u)g(r(u)_x)_x)_x = 0, \quad a < x < b,\ t > 0 \tag{2.1}$$

with the initial condition

$$u(x,0) = u_0(x), \quad a \leq x \leq b$$

where $f(u), r(u)$, and $g(r)$ are given smooth functions. In this work, we consider a uniform mesh of cell size Δx. Denote the cells by $I_j = (x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}})$, where $x_{j+\frac{1}{2}} = \frac{1}{2}(x_j + x_{j+1})$. As for the HWENO scheme [14,15], we want to find the approximations for the cell averages of u and $u\frac{x-x_j}{\Delta x}$ of the solution of (2.1), i.e.,

$$\bar{u}_j \approx \frac{1}{\Delta x}\int_{I_j} u dx, \quad \bar{v}_j \approx \frac{1}{\Delta x}\int_{I_j} u\frac{x-x_j}{\Delta x}dx.$$

Similar to the procedure of HWENO limiter for DG in [14,15], we use here the first moment $\bar{v}_j$ instead of the cell average of u_x which was used for HWENO scheme in [14,15]. This is more consistent with the basis functions of LDG (see (2.5) below). We employ LDG (e.g., see [21,23]) for the discretization of high order spatial derivatives in (2.1). We first introduce the new variables

$$q = r(u)_x, \qquad p = g(q)_x,$$

and rewrite (2.1) as a system of first order differential equations,

$$u_t + (f(u) + r'(u)p)_x = 0, \tag{2.2}$$

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

$$q = r(u)_x. \tag{2.4}$$

Next, we consider the discretization of (2.3) and (2.4). Let p and q be approximated by

$$p_h = \sum_{l=0}^{k} p_j^{(l)}\phi_l^{(j)}(x), \quad q_h = \sum_{l=0}^{k} q_j^{(l)}\phi_l^{(j)}(x), \quad \forall x \in I_j$$

where $\phi_l^{(j)}(x)$'s are the orthogonal basis functions and k is the highest degree of polynomials used in the basis functions. The first five basis functions are

$$\begin{aligned}
&\phi_0^{(j)}(x)=1, && \phi_1^{(j)}(x)=\frac{x-x_j}{\Delta x}, \\
&\phi_2^{(j)}(x)=(\frac{x-x_j}{\Delta x})^2-\frac{1}{12}, && \phi_3^{(j)}(x)=(\frac{x-x_j}{\Delta x})^3-\frac{3}{20}\frac{x-x_j}{\Delta x}, \\
&\phi_4^{(j)}(x)=(\frac{x-x_j}{\Delta x})^4-\frac{3}{14}(\frac{x-x_j}{\Delta x})^2+\frac{3}{560}.
\end{aligned} \tag{2.5}$$

Multiplying (2.3) and (2.4) by test functions w and z, respectively, integrating over the interval I_j, integrating by parts, and replacing function values at cell boundaries by their numerical fluxes (those quantities with "hats") which we will define in the below, we obtain

$$\int_{I_j} p_h w dx+\int_{I_j} g(q_h) w_x dx-\hat{g}_{j+\frac12}w^-_{j+\frac12}+\hat{g}_{j-\frac12}w^+_{j-\frac12}=0, \tag{2.6}$$

$$\int_{I_j} q_h z dx+\int_{I_j} r(u) z_x dx-\hat{r}_{j+\frac12}z^-_{j+\frac12}+\hat{r}_{j-\frac12}z^+_{j-\frac12}=0, \tag{2.7}$$

where

$$w^\pm_{j\pm\frac12}=\lim_{x\to x_{j\pm\frac12}\pm 0} w(x),\quad z^\pm_{j\pm\frac12}=\lim_{x\to x_{j\pm\frac12}\pm 0} z(x).$$

The test functions w and z in (2.6) and (2.7) are taken as the basis functions $\phi_l^{(j)}(x)$, $l=0,1,\cdots,k$, successively. The integrals in (2.6) and (2.7) are computed numerically using a Gaussian quadrature rule. For the $(k+1)$th order accuracy, the integration must achieve $(2k+2)$th order as the DG method. Thus, we use the $(k+1)$-point Gaussian quadrature, namely,

$$\int_{I_j} g(q_h) w_x dx\approx \Delta x\sum_G g(q_h(x_G))w_x(x_G)w_G,$$

$$\int_{I_j} r(u) z_x dx\approx \Delta x\sum_G r(u(x_G))z_x(x_G)w_G,$$

where w_G's are the weights and x_G's are the Gauss points. The function q_h used in the first integral is obtained by solving (2.7) while the value $u(x_G)$ used in the second integral is reconstructed from the cell averages $\{\bar{u}_j,\bar{v}_j\}$. For example, the sixth order reconstruction procedure includes following three steps:

Step 1. Define the small stencils

$$S_0=\{I_{j-1},I_j,I'_{j-1},I'_j\},\quad S_1=\{I_j,I_{j+1},I'_j,I'_{j+1}\},\quad S_2=\{I_{j-1},I_j,I_{j+1},I'_j\}$$

and a bigger stencil $S=\{S_0,S_1,S_2\}$, where I_j and I'_j stand for intervals chosen for $\bar{u}_j$ and $\bar{v}_j$, respectively. These stencils are used to construct polynomials $p_0(x)$, $p_1(x)$, and $p_2(x)$ of degree three and polynomial $Q(x)$ of degree five,

$$\frac{1}{\Delta x}\int_{I_{j+l}} p_0(x)dx=\bar{u}_{j+l},\quad \frac{1}{\Delta x}\int_{I'_{j+l}=I_{j+l}} p_0(x)\frac{x-x_j}{\Delta x}dx=\bar{v}_{j+l},\quad l=-1,0$$

$$\frac{1}{\Delta x}\int_{I_{j+l}} p_1(x)dx=\bar{u}_{j+l},\quad \frac{1}{\Delta x}\int_{I'_{j+l}=I_{j+l}} p_1(x)\frac{x-x_j}{\Delta x}dx=\bar{v}_{j+l},\quad l=0,1$$

$$\frac{1}{\Delta x}\int_{I_{j+l}} p_2(x)dx=\bar{u}_{j+l},\quad \frac{1}{\Delta x}\int_{I'_j=I_j} p_2(x)\frac{x-x_j}{\Delta x}dx=\bar{v}_j,\quad l=-1,0,1$$

$$\frac{1}{\Delta x}\int_{I_{j+l}} Q(x)dx=\bar{u}_{j+l},\quad \frac{1}{\Delta x}\int_{I'_{j+l}=I_{j+l}} Q(x)\frac{x-x_j}{\Delta x}dx=\bar{v}_{j+l},\quad l=-1,0,1.$$

Step 2. For a given point $\hat{x}\in I_j$, we find the linear weights, denoted by γ_0,γ_1 and γ_2, such that

$$Q(\hat{x})=\sum_{i=0}^{2}\gamma_i p_i(\hat{x}).$$

For example, for $\hat{x} = x_{j+\frac{1}{2}}$ we have

$$\gamma_0 = \frac{25}{189}, \quad \gamma_1 = \frac{14}{27}, \quad \gamma_2 = \frac{22}{63}.$$

Step 3. Finally, the HWENO reconstruction is given by

$$u(\hat{x}) = \sum_{i=0}^{2} \gamma_i p_i(\hat{x}) = Q(\hat{x}). \tag{2.8}$$

Step 2 to Step 3 are repeated for each point $\hat{x} \in \{\{x_G\}, x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}\}$, where $\{x_G\}$ denotes the set of the Gauss points. Note that we have used only linear weights in the above reconstruction procedure (i.e., we only use reconstruction polynomial $Q(x)$ in big stencil) since the solutions of KdV-type equations are smooth in general.

It is remarked that the reconstructed values of u at both ends of I_j are denoted by $u^+_{j-\frac{1}{2}}$ and $u^-_{j+\frac{1}{2}}$, respectively. Moreover, the linear weights for some points can become negative. For example, for $x_G = x_{j-\frac{0.538469310105683l}{2}}$, the linear weights are $\gamma_0 = -1.19876833424689$, $\gamma_1 = -0.189130224626382$, and $\gamma_2 = 2.38789855887328$. The negative weights may lead to instability of the reconstruction. The technique developed by Shi et al. [17] can be used to treat reconstruction procedure with negative weights; the interested reader is referred to [17] for the detail.

Multiplying (2.2) with 1 and $\frac{x-x_j}{\Delta x}$ and integrating them by parts over I_j, we have

$$\frac{d\bar{u}_j}{dt} = -\frac{1}{\Delta x}(f(u) + r'(u)p)|_{I_j}, \tag{2.9}$$

$$\frac{d\bar{v}_j}{dt} = -\frac{1}{\Delta x}(f(u) + r'(u)p)\frac{x-x_j}{\Delta x}|_{I_j} + \frac{1}{\Delta x^2}\int_{I_j}(f(u) + r'(u)p)dx. \tag{2.10}$$

The integral term in (2.10) is approximated by the $(k+1)$-point Gaussian quadrature rule as for (2.6) and (2.7). We obtain the numerical scheme as

$$\frac{d\bar{u}_j}{dt} = -\frac{1}{\Delta x}[(\hat{f}_{j+\frac{1}{2}} + \hat{r}'_{j+\frac{1}{2}}\hat{p}_{j+\frac{1}{2}}) - (\hat{f}_{j-\frac{1}{2}} + \hat{r}'_{j-\frac{1}{2}}\hat{p}_{j-\frac{1}{2}})], \tag{2.11}$$

$$\frac{d\bar{v}_j}{dt} = -\frac{1}{2\Delta x}[(\hat{f}_{j+\frac{1}{2}} + \hat{r}'_{j+\frac{1}{2}}\hat{p}_{j+\frac{1}{2}}) + (\hat{f}_{j-\frac{1}{2}} + \hat{r}'_{j-\frac{1}{2}}\hat{p}_{j-\frac{1}{2}})] + \frac{1}{\Delta x^2}F_j, \tag{2.12}$$

where

$$F_j = \Delta x \sum_G (f(u(x_G)) + r'(u(x_G))p_h(x_G))w_G \approx \int_{I_j}(f(u) + r'(u)p_h)dx.$$

A key component of the above described hybrid LDG-HWENO scheme is to define the numerical fluxes in (2.6), (2.7), (2.11), and (2.12) to ensure the accuracy and stability of the scheme. We define

$$\begin{aligned}
\hat{f}_{j+\frac{1}{2}} &= \hat{f}(u^-_{j+\frac{1}{2}}, u^+_{j+\frac{1}{2}}), & \hat{g}_{j+\frac{1}{2}} &= \hat{g}(q^-_{j+\frac{1}{2}}, q^+_{j+\frac{1}{2}}), \\
\hat{p}_{j+\frac{1}{2}} &= p^+_{j+\frac{1}{2}}, & \hat{r}_{j+\frac{1}{2}} &= r(u^-_{j+\frac{1}{2}}), \\
\hat{r}'_{j+\frac{1}{2}} &= \frac{r(u^+_{j+\frac{1}{2}}) - r(u^-_{j+\frac{1}{2}})}{u^+_{j+\frac{1}{2}} - u^-_{j+\frac{1}{2}}},
\end{aligned}$$

where

$$p^\pm_{j+\frac{1}{2}} = p_h(x_{j+\frac{1}{2}} \pm 0), \quad q^\pm_{j+\frac{1}{2}} = q_h(x_{j+\frac{1}{2}} \pm 0),$$

and $u^\pm_{j+\frac{1}{2}}$ are the reconstructed values of u by the HWENO method as described in Step 1–Step 5. It is noted that the choice of the numerical fluxes is not unique. The key part is that $\hat{p}$ and $\hat{r}$ must be taken from the opposite sides. The numerical flux $\hat{f}(a,b)$ for $f(u)$ should satisfy the following conditions:

(i) $\hat{f}(a,b)$ is a Lipschitz continuous function in both arguments a and b;
(ii) $\hat{f}$ is consistent with $f(u)$, namely, $\hat{f}(u,u) = f(u)$;
(iii) $\hat{f}(a,b)$ is a monotone flux, i.e., it is non-decreasing in a and non-increasing in b.

In this paper, we use the local Lax–Friedrichs (LLF) flux,

$$\hat{f}(a,b) = \frac{1}{2}[f(a) + f(b) - \alpha(b-a)], \quad \alpha = \max_{u\in[\min(a,b),\max(a,b)]} |f'(u)|.$$

The numerical flux $\hat{g}(c,d)$ for $g(q)$ should also satisfy conditions (i) and (ii) but the condition (iii) should be replaced by

(iii)′ $-\hat{g}(q^-, q^+)$ is a monotone flux for $-g(q)$, namely, $\hat{g}(c,d)$ is non-increasing in c and non-decreasing in d.

Corresponding to the LLF flux, we have

$$\hat{g}(c,d) = \frac{1}{2}[g(c) + g(d) - \alpha(c-d)], \quad \alpha = \max_{q\in[\min(c,d),\max(c,d)]} |g'(q)|.$$

We note that (2.7) can be solved independently on each cell for q_h in terms of the cell averages $\{\bar{u}_j, \bar{v}_j\}$. Similarly, (2.6) can be solved on each cell for p_h in terms of q_h. As a consequence, the global unknown variables involve only $\bar{u}_j$ and $\bar{v}_j$, which are governed by a system of ordinary differential equations. In our computation, the system is integrated in time using an explicit TVD Runge–Kutta method [18], such as the 3rd TVD Runge–Kutta method for solving

$$u_t = L(u,t),$$

where $L(u,t)$ is a spatial discretization operator,

$$\begin{aligned} u^* &= u^n + \Delta t L(u^n, t^n) \\ u^{**} &= \frac{3}{4}u^n + \frac{1}{4}(u^* + \Delta t L(u^*, t^n + \Delta t)) \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}(u^{**} + \Delta t L(u^{**}, t^n + \frac{1}{2}\Delta t)). \end{aligned}$$

From the construction of the scheme, it is not difficult to show formally that the method is of $(k+1)$th order in space ($k \le 4$) and third order in time, i.e.,

$$e_h^n = \mathcal{O}(\Delta t^3) + \mathcal{O}(\Delta x^{k+1}).$$

Since an explicit scheme is used, the time step is subject to the CFL condition and should be taken as $\Delta t = \mathcal{O}(\Delta x^3)$. For this choice, the error is dominated by the spatial discretization error.

3. The hybrid LDG-HWENO method in two dimensions

In this section we describe the method for two dimensional problems. We consider the general form

$$\begin{aligned} &u_t + f_1(u)_x + f_2(u)_y + [r_1'(u)(g_{11}(r_1(u)_x)_x + g_{12}(r_1(u)_x)_y)]_x \\ &+ [r_2'(u)(g_{21}(r_2(u)_y)_x + g_{22}(r_2(u)_y)_y)]_y = 0, \quad (x,y) \in (a,b)\times(c,d) \end{aligned} \tag{3.1}$$

subject to a periodic boundary condition and the initial condition

$$u(x,y,0) = u_0(x,y), \quad (x,y) \in (a,b)\times(c,d)$$

where $f_m(u)$, $r_m(u)$, and $g_{mn}(r)$ are given smooth nonlinear functions.

We use a rectangle mesh of cell size Δx and Δy in x and y directions, respectively. We denote the cells by

$$I_{ij} = (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}) \times (y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}),$$

where

$$x_{i+\frac{1}{2}} = \frac{1}{2}(x_i + x_{i+1}), \quad y_{j+\frac{1}{2}} = \frac{1}{2}(y_j + y_{j+1}).$$

We approximate the cell averages of u, $u\frac{x-x_i}{\Delta x}$, $u\frac{y-y_j}{\Delta y}$ and $u\frac{x-x_i}{\Delta x}\frac{y-y_j}{\Delta y}$ by

$$\bar{u}_{ij} \approx \frac{1}{\Delta x \Delta y}\int_{I_{ij}} u dxdy,$$

$$\bar{v}_{ij} \approx \frac{1}{\Delta x \Delta y}\int_{I_{ij}} u\frac{x-x_i}{\Delta x} dxdy,$$

$$\bar{w}_{ij} \approx \frac{1}{\Delta x \Delta y} \int_{I_{ij}} u \frac{y - y_j}{\Delta y} dxdy,$$

$$\bar{z}_{ij} \approx \frac{1}{\Delta x \Delta y} \int_{I_{ij}} u \frac{x - x_i}{\Delta x} \frac{y - y_j}{\Delta y} dxdy.$$

For the discretization of high order spatial derivatives using LDG, we introduce the new variables

$$q_1 = r_1(u)_x, \quad q_2 = r_2(u)_y, \quad p_1 = g_{11}(q_1)_x + g_{12}(q_1)_y, \quad p_2 = g_{21}(q_2)_x + g_{22}(q_2)_y, \tag{3.2}$$

and rewrite (3.1) as a first order system

$$u_t + (f_1(u) + r_1'(u)p_1)_x + (f_2(u) + r_2'(u)p_2)_y = 0, \tag{3.3}$$

coupled with (3.2). We approximate $p_m, q_m (m = 1, 2)$ by polynomials,

$$p_m = \sum_{l=0}^{k} p_{m,l}^{(ij)} \phi_l^{(ij)}(x, y), \quad q_m = \sum_{l=0}^{k} q_{m,l}^{(ij)} \phi_l^{(ij)}(x, y), \quad m = 1, 2 \quad \forall (x, y) \in I_{ij}$$

where $\phi_l^{(ij)}(x, y)$'s are the basis functions,

$$\phi_0^{(ij)} = 1, \quad \phi_1^{(ij)} = \frac{x - x_i}{\Delta x}, \quad \phi_2^{(ij)} = \frac{y - y_j}{\Delta y}, \quad \phi_3^{(ij)} = (\frac{x - x_i}{\Delta x})^2 - \frac{1}{12}$$

$$\phi_4^{(ij)} = (\frac{x - x_i}{\Delta x})(\frac{y - y_j}{\Delta y}), \quad \phi_5^{(ij)} = (\frac{y - y_j}{\Delta y})^2 - \frac{1}{12}, \quad \cdots$$

To discretize (3.2), we multiply it by test functions w_m and z_m, respectively, and integrate the resulting equations by parts over the cell I_{ij}. Using numerical fluxes for quantities on the boundary of the cell, we have

$$\begin{aligned} &\int_{I_{ij}} p_1 w_1 dxdy + \int_{I_{ij}} g_{11}(q_1)(w_1)_x dxdy + \int_{I_{ij}} g_{12}(q_1)(w_1)_y dxdy \\ &- \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \left[\hat{g}_{11}(q_1(x_{i+\frac{1}{2}}, y)) w_1(x_{i+\frac{1}{2}}^-, y) - \hat{g}_{11}(q_1(x_{i-\frac{1}{2}}, y)) w_1(x_{i-\frac{1}{2}}^+, y) \right] dy \\ &- \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left[\hat{g}_{12}(q_1(x, y_{j+\frac{1}{2}})) w_1(x, y_{j+\frac{1}{2}}^-) - \hat{g}_{12}(q_1(x, y_{j-\frac{1}{2}})) w_1(x, y_{j-\frac{1}{2}}^+) \right] dx = 0, \end{aligned} \tag{3.4}$$

$$\begin{aligned} &\int_{I_{ij}} p_2 w_2 dxdy + \int_{I_{ij}} g_{21}(q_2)(w_2)_x dxdy + \int_{I_{ij}} g_{22}(q_2)(w_2)_y dxdy \\ &- \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \left[\hat{g}_{21}(q_2(x_{i+\frac{1}{2}}, y)) w_2(x_{i+\frac{1}{2}}^-, y) - \hat{g}_{21}(q_2(x_{i-\frac{1}{2}}, y)) w_2(x_{i-\frac{1}{2}}^+, y) \right] dy \\ &- \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left[\hat{g}_{22}(q_2(x, y_{j+\frac{1}{2}})) w_2(x, y_{j+\frac{1}{2}}^-) - \hat{g}_{22}(q_2(x, y_{j-\frac{1}{2}})) w_2(x, y_{j-\frac{1}{2}}^+) \right] dx = 0, \end{aligned} \tag{3.5}$$

$$\begin{aligned} &\int_{I_{ij}} q_1 z_1 dxdy + \int_{I_{ij}} r_1(u)(z_1)_x dxdy \\ &- \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \left[\hat{r}_1(u(x_{i+\frac{1}{2}}, y)) z_1(x_{i+\frac{1}{2}}^-, y) - \hat{r}_1(u(x_{i-\frac{1}{2}}, y)) z_1(x_{i-\frac{1}{2}}^+, y) \right] dy = 0, \end{aligned} \tag{3.6}$$

$$\int\limits_{I_{ij}} q_2 z_2 dxdy + \int\limits_{I_{ij}} r_2(u)(z_2)_y dxdy$$
$$- \int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left[\hat{r}_2(u(x, y_{j+\frac{1}{2}}))z_2(x, y^-_{j+\frac{1}{2}}) - \hat{r}_2(u(x, y_{j-\frac{1}{2}}))z_2(x, y^+_{j-\frac{1}{2}})\right] dx = 0. \tag{3.7}$$

This leads to equations including double integrals and line integrals on the cell I_{ij}. As for the one dimensional case, the volume integrals can be approximated by the tensor product Gaussian quadrature rule with $(k+1)$ points in each direction, and the line integrals can be computed by the $(k+1)$-point Gaussian quadrature rule.

The final discrete equations are obtained by replacing those the volume and line integrals by the Gaussian quadrature formulas. To save space, we omit these equations here. Instead, we point out that p_1, p_2, q_1, q_2 are obtained by solving (3.4)–(3.7) and the values of u at $(\hat{x}, \hat{y})$ with $\hat{x} \in \{x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}, \{x_{G_1}\}\}$ and $\hat{y} \in \{y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}, \{y_{G_2}\}\}$ are reconstructed from the cell averages as in the one dimensional case. For reconstruction on Cartesian meshes, one can adopt either a direct two dimensional procedure or a dimension-by-dimension strategy [17]. In this paper, we use the dimension-by-dimension strategy [24]. First, we perform two y-direction reconstructions, i.e.,

$$\{\bar{u}_{mn}, \bar{w}_{mn}\} \longrightarrow \bar{u}_{i+l,j}(\hat{y}) \approx \frac{1}{\Delta x} \int\limits_{I_{i+l,j}} u(x, \hat{y}) dx, \quad l = -1, 0, 1, \quad \hat{y} \in \{y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}, \{y_{G_2}\}\}$$

$$\{\bar{v}_{mn}, \bar{Z}_{mn}\} \longrightarrow \bar{u}_{x,i+l,j}(\hat{y}) \approx \frac{1}{\Delta x} \int\limits_{I_{i+l,j}} u_x(x, \hat{y}) dx, \quad l = -1, 0, 1, \quad \hat{y} \in \{y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}, \{y_{G_2}\}\}.$$

Then we use $\bar{u}(\hat{y})$ and $\bar{u}_x(\hat{y})$ to perform x-direction reconstruction to get an approximation to $u(\hat{x}, \hat{y})$, i.e.,

$$\{\bar{u}_{mn}(\hat{y}), \bar{u}_{x,m,n}(\hat{y})\} \longrightarrow \widetilde{u}(\hat{x}, \hat{y}) \approx u(\hat{x}, \hat{y}), \quad \hat{x} \in \{x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}, \{x_{G_1}\}\}, \quad \hat{y} \in \{y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}, \{y_{G_2}\}\}.$$

The values $\widetilde{u}(\hat{x}, \hat{y})$ are used in computing the volume and line integrals.

Multiplying (3.3) with 1, $\frac{x-x_i}{\Delta x}$, $\frac{y-y_j}{\Delta y}$ and $\frac{x-x_i}{\Delta x}\frac{y-y_j}{\Delta y}$, integrating over I_{ij} by parts, and using numerical fluxes on the cell boundary, we get

$$\frac{d\bar{u}_{ij}}{dt} = -\frac{1}{\Delta x \Delta y} \int\limits_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \hat{H}_1(x_{i+\frac{1}{2}}, y) - \hat{H}_1(x_{i-\frac{1}{2}}, y) dy$$
$$- \frac{1}{\Delta x \Delta y} \int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \hat{H}_2(x, y_{j+\frac{1}{2}}) - \hat{H}_2(x, y_{j-\frac{1}{2}}) dx, \tag{3.8}$$

$$\frac{d\bar{v}_{ij}}{dt} = -\frac{1}{2\Delta x \Delta y} \int\limits_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \hat{H}_1(x_{i+\frac{1}{2}}, y) + \hat{H}_1(x_{i-\frac{1}{2}}, y) dy$$
$$- \frac{1}{\Delta x \Delta y} \int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} (\hat{H}_2(x, y_{j+\frac{1}{2}}) - \hat{H}_2(x, y_{j-\frac{1}{2}})) \frac{x - x_i}{\Delta x} dx$$
$$+ \frac{1}{\Delta x^2 \Delta y} \int\limits_{I_{ij}} H_1 dxdy, \tag{3.9}$$

$$\frac{d\bar{w}_{ij}}{dt} = -\frac{1}{\Delta x \Delta y} \int\limits_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} (\hat{H}_1(x_{i+\frac{1}{2}}, y) - \hat{H}_1(x_{i-\frac{1}{2}}, y)) \frac{y - y_j}{\Delta y} dy$$

$$
\begin{aligned}
&-\frac{1}{2\Delta x\Delta y}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}(\hat{H}_2(x,y_{j+\frac{1}{2}})+\hat{H}_2(x,y_{j-\frac{1}{2}}))dx\\
&+\frac{1}{\Delta x\Delta y^2}\int_{I_{ij}}H_2dxdy, \qquad (3.10)
\end{aligned}
$$

$$
\begin{aligned}
\frac{d\bar{Z}_{ij}}{dt}=&-\frac{1}{2\Delta x\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}}(\hat{H}_1(x_{i+\frac{1}{2}},y)+\hat{H}_1(x_{i-\frac{1}{2}},y))\frac{y-y_j}{\Delta y}dy\\
&-\frac{1}{2\Delta x\Delta y}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}(\hat{H}_2(x,y_{j+\frac{1}{2}})+\hat{H}_2(x,y_{j-\frac{1}{2}}))\frac{x-x_i}{\Delta x}dx\\
&+\frac{1}{\Delta x^2\Delta y}\int_{I_{ij}}H_1\frac{y-y_j}{\Delta y}dxdy+\frac{1}{\Delta x\Delta y^2}\int_{I_{ij}}H_2\frac{x-x_i}{\Delta x}dxdy, \qquad (3.11)
\end{aligned}
$$

where $H_1 = f_1(u) + r_1'(u)p_1$ and $H_2 = f_2(u) + r_2'(u)p_2$.

The volume integrals and line integrals are approximated by a Gaussian quadrature rule as for (3.4)–(3.7). If we use G_b to stand for a boundary point $(\hat{x}, \hat{y})$, then the numerical fluxes $\hat{H}_i(G_b)$, $\hat{r}_i(G_b)$, and $\hat{g}_{ij}(G_b)$ $(i = 1, 2,\ j = 1, 2)$ can be defined similarly as in one dimension, viz.,

$$
\begin{aligned}
&\hat{f}_i(u(G_b))=\hat{f}_i(u^-(G_b),u^+(G_b)), && \hat{g}_{ij}=\hat{g}_{ij}(q_i^-(G_b),q_i^+(G_b)),\\
&\hat{p}_i=p_i^+(G_b), && \hat{r}_i=r_i(u^-(G_b)),\\
&\hat{r}_i'(u(G_b))=\frac{r_i(u^+(G_b))-r_i(u^-(G_b))}{u^+(G_b)-u^-(G_b)}, && \hat{H}_i=\hat{f}_i+\hat{r}_i'\hat{p}_i,
\end{aligned}
$$

where $u^{\pm}(G_b)$, $q_i^{\pm}(G_b)$, and $p_i^{\pm}(G_b)$ are the left (or "in") and right (or "out") limits of the solutions u, q_i, and p_i at the cell interface G_b, respectively. As in one dimension, $\hat{f}_i(u(G_b))$ is a monotone flux for $\hat{f}_i(u, u) = f_i(u)$ and $-\hat{g}_{ij}$ is a flux for $-\hat{g}_{ij}(q_i, q_i) = -g_{ij}(q_i)$. Also, the choice of the fluxes is not unique. We must take $\hat{p}$ and $\hat{r}$ from the opposite sides.

Once again, the discrete equations can be obtained accordingly. To save space, they are omitted here.

The resultant ODE system is integrated in time with an explicit third order TVD Runge–Kutta method.

4. Numerical examples

In this section we present numerical results obtained with the hybrid LDG-HWENO scheme described in the previous sections for four examples each in one and two dimensions.

4.1. One dimensional examples

Example 4.1. We compute the solution of the linear equation

$$u_t + u_{xxx} = 0, \quad 0 < x < 2\pi,\ t > 0$$

subject to the initial condition $u(x, 0) = \sin(x)$ and the periodic boundary condition. The equation is in the form of (2.1) with $f(u) = 0$, $r(u) = u$, $g(r) = r$. The exact solution is given by $u(x,t) = \sin(x + t)$. The L^1, L^2, and L^∞ norm of the error and the convergence order are shown in Table 4.1. It can be seen that the scheme with P^k elements in DG gives at least a $(k + 1)$th order of accuracy.

Recall that the explicit 3rd TVD Runge–Kutta method is employed for time integration. The CFL stability condition requires the time step to satisfy

$$\Delta t \le C\Delta x^3 \qquad (4.1)$$

for some constant C depending on the PDE under consideration and the highest degree (k) of approximation polynomials but not on Δt and Δx. For the current example, it is found that $C \approx 0.00909$, 0.00149, and 0.00147 for $k = 2$, 3, and 4, respectively. Apparently, C decreases with increasing k.

Table 4.1
Example 4.1: periodic boundary conditions, and $T = 1.0$.

k	n	10	20	40	80	160
2	L^1	1.418e−3	1.540e−4	1.849e−5	2.275e−6	2.826e−7
	Order		3.20	3.06	3.02	3.01
	L^2	1.465e−3	1.669e−4	2.023e−5	2.507e−6	3.127e−7
	Order		3.13	3.04	3.01	3.00
	L_∞	1.942e−3	2.284e−4	2.812e−5	3.513e−6	4.403e−7
	Order		3.09	3.02	3.00	3.00
3	L^1	4.701e−5	2.545e−6	1.511e−7	9.275e−9	5.770e−10
	Order		4.21	4.07	4.03	4.01
	L^2	5.121e−5	2.785e−6	1.661e−7	1.024e−8	6.389e−10
	Order		4.20	4.07	4.02	4.00
	L_∞	7.113e−5	3.887e−6	2.341e−7	1.444e−8	9.021e−10
	Order		4.19	4.05	4.02	4.00
4	L^1	1.174e−4	3.655e−6	1.145e−7	3.618e−9	1.110e−10
	Order		5.01	5.00	4.98	5.03
	L^2	1.217e−4	3.961e−6	1.252e−7	3.988e−8	1.228e−10
	Order		4.94	4.98	4.97	5.02
	L_∞	1.571e−4	5.359e−6	1.734e−7	5.584e−9	1.728e−10
	Order		4.87	4.95	4.96	5.01

Table 4.2
Example 4.2: periodic boundary conditions and $T = 0.5$.

k	n	40	80	160	320	640
2	L^1	6.218e−3	7.157e−4	8.185e−5	9.755e−6	1.198e−6
	Order		3.12	3.13	3.07	3.03
	L^2	9.789e−3	1.084e−3	1.243e−4	1.488e−5	1.831e−6
	Order		3.17	3.12	3.06	3.02
	L_∞	3.793e−2	4.225e−3	4.752e−4	5.716e−5	7.060e−6
	Order		3.17	3.15	3.06	3.02
3	L^1	2.318e−3	5.273e−5	2.076e−6	1.064e−7	6.606e−9
	Order		5.46	4.67	4.29	4.01
	L^2	3.145e−3	7.757e−5	3.237e−6	1.689e−7	1.019e−8
	Order		5.34	4.58	4.26	4.05
	L_∞	8.637e−3	2.799e−4	1.095e−5	6.143e−7	3.836e−8
	Order		4.95	4.68	4.16	4.00
4	L^1	5.904e−3	8.765e−5	3.265e−6	1.119e−7	4.293e−9
	Order		6.07	4.75	4.87	4.70
	L^2	7.852e−3	1.299e−4	4.637e−6	1.591e−7	5.851e−9
	Order		5.92	4.81	4.87	4.77
	L_∞	2.768e−2	6.381e−4	2.078e−5	6.978e−7	2.436e−8
	Order		5.44	4.94	4.90	4.84

Example 4.2. In order to see the accuracy of the scheme for nonlinear problems, we compute the classical soliton solution of the KdV equation

$$u_t - 3(u^2)_x + u_{xxx} = 0, \quad -10 \le x \le 12.$$

The initial condition is given by

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

and the exact solution is

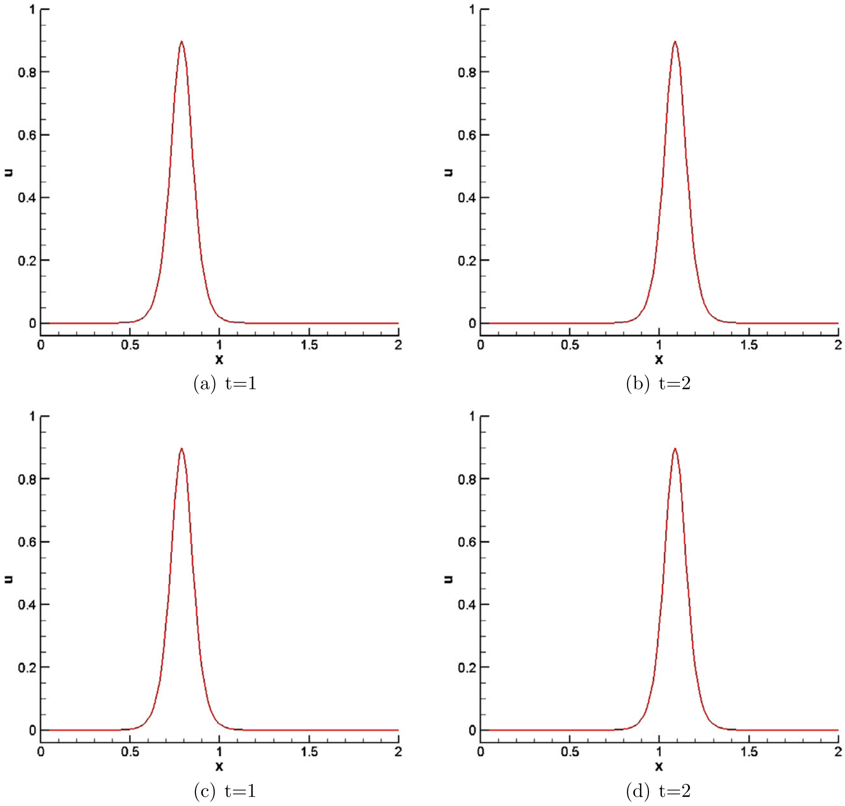
$$u(x, t) = -2\text{sech}^2(x - 4t).$$

For this example, $f(u) = -3u^2$, $r(u) = u$, $g(r) = r$. Table 4.2 gives the error of the numerical solution at $t = 0.5$. We can see that the $(k + 1)$th order of accuracy of the scheme is achieved for this nonlinear problem.

Example 4.3. In this example we compute several classical soliton solutions of the KdV equation

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

Here, $f(u) = \frac{u^2}{2}$, $r(u) = u$, $g(r) = \epsilon r$.



(a) t=1

(b) t=2

(c) t=1

(d) t=2

Fig. 4.1. Example 4.3: Single soliton case. Top: P^2 elements are used with 160 cells; Bottom: P^4 elements are used with 160 cells.

The single soliton case has the initial condition

$$u(x,0) = 3c\,\mathrm{sech}^2(k(x - x_0))$$

with $c = 0.3$, $x_0 = 0.5$, $k = \frac{1}{2}\sqrt{\frac{c}{\epsilon}}$, and $\epsilon = 5 \times 10^{-4}$. The solution is computed in $x \in (0, 2)$ with periodic boundary conditions and shown in Fig. 4.1.

The double soliton collision case has the initial condition

$$u(x,0) = 3c_1\mathrm{sech}^2(k_1(x - x_1)) + 3c_2\mathrm{sech}^2(k_2(x - y))$$

with $c_1 = 0.3$, $c_2 = 0.1$, $x_1 = 0.4$, $y = 0.8$, $k_i = \frac{1}{2}\sqrt{\frac{c_i}{\epsilon}}$ for $i = 1, 2$, and $\epsilon = 4.84 \times 10^{-4}$. The solution shown in Fig. 4.2 is computed in $x \in (0, 2)$ with a periodic boundary condition.

The triple soliton splitting case has the initial condition

$$u(x,0) = \frac{2}{3}\mathrm{sech}^2(\frac{x-1}{\sqrt{108\epsilon}})$$

with $\epsilon = 10^{-4}$. The solution shown in Fig. 4.3 is computed in $x \in (0, 3)$ with a periodic boundary condition.

In order to see the long time performance of the scheme, we compute all the solitons up to $t = 200$. The numerical solutions are shown in Figs. 4.4–4.6. It can be seen that the peaks of the solitons are almost the same (except for the interaction regions). This indicates that the new scheme performs well even for a relatively long time.

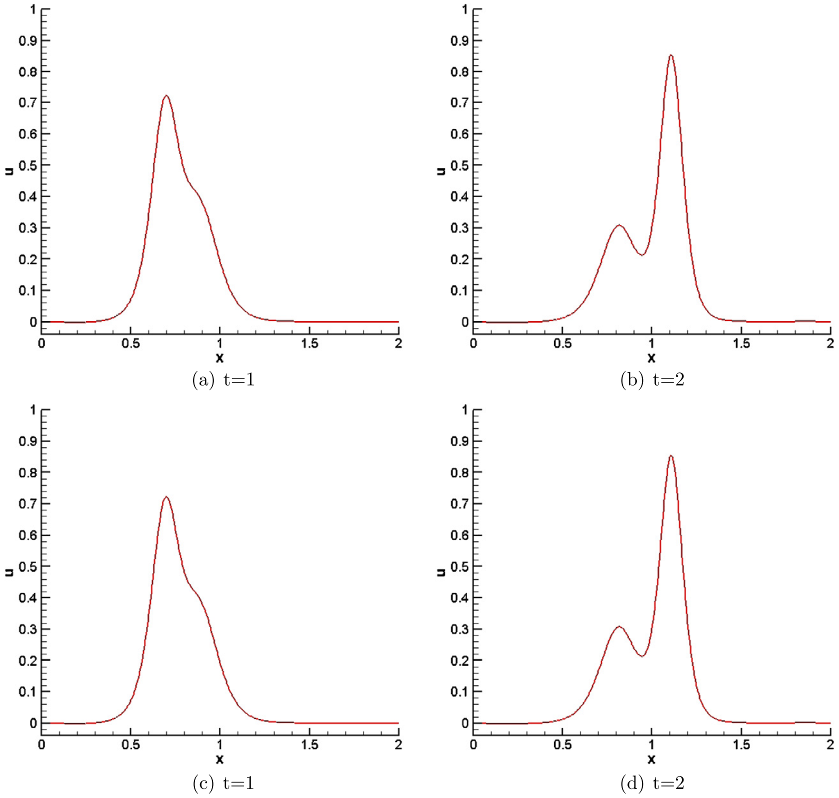
Example 4.4. We compute in this example the KdV zero dispersion limit of conservation laws. The equation is

$$u_t + (\frac{u^2}{2})_x + \epsilon u_{xxx} = 0, \quad x \in (0, 1)$$

subject to the periodic boundary condition and the initial condition

$$u(x,0) = 2 + 0.5\sin(2\pi x).$$

We compute the solution to $t = 0.5$ with $\epsilon = 10^{-4}$, 10^{-5}, 10^{-6}, and 10^{-7}. These numerical solutions are shown in Figs. 4.7 and 4.8.



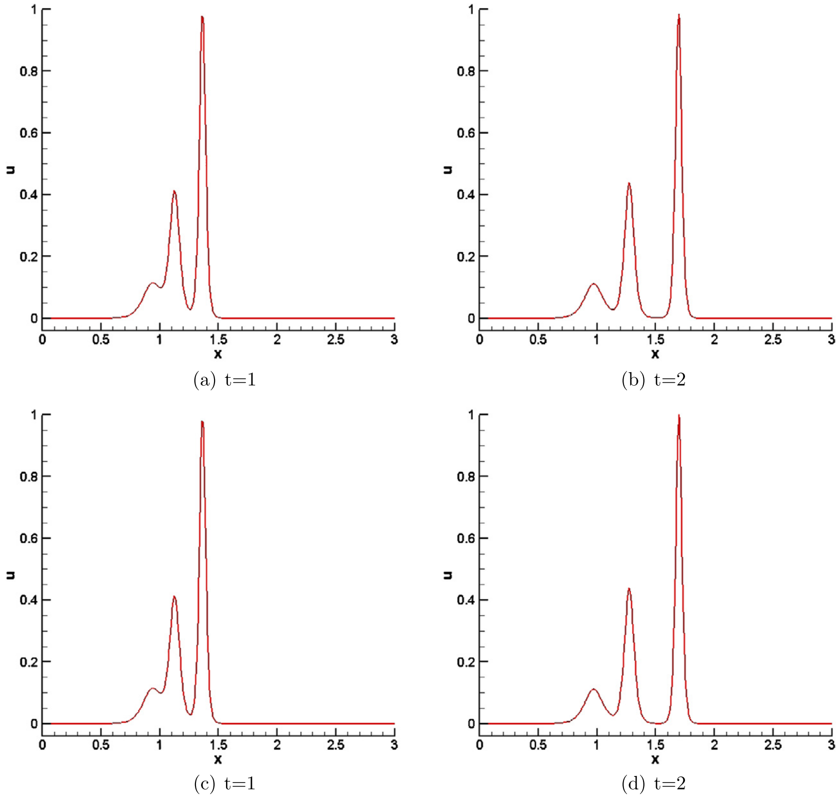
(a) t=1

(b) t=2

(c) t=1

(d) t=2

Fig. 4.2. Example 4.3: Double soliton collision case. Top: P^2 elements are used with 320 cells; Bottom: P^4 elements are used with 320 cells.



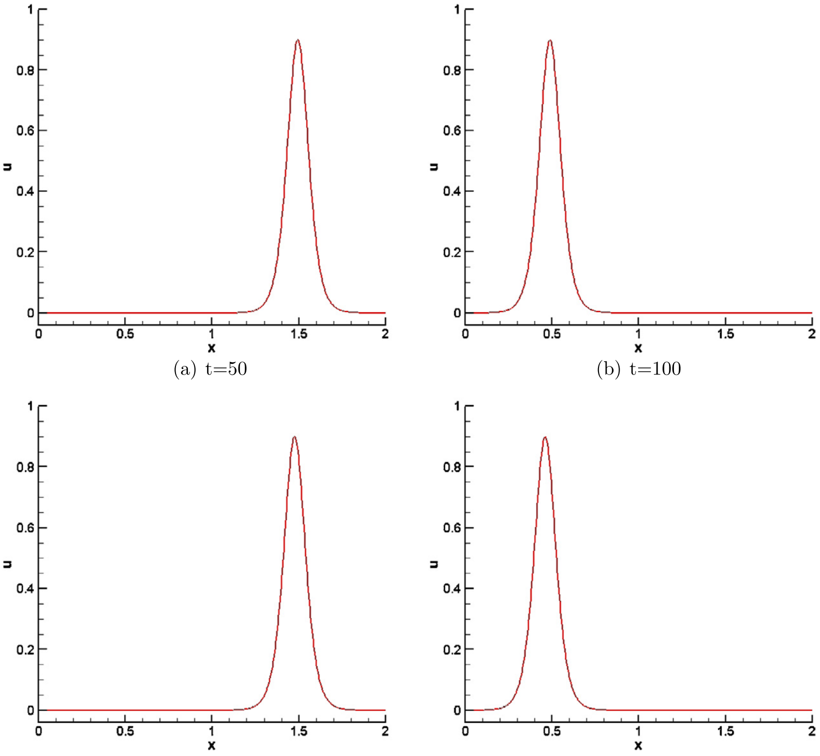
(a) t=1

(b) t=2

(c) t=1

(d) t=2

Fig. 4.3. Example 4.3: Triple soliton splitting case. Top: P^2 elements are used with 320 cells; Bottom: P^4 elements are used with 320 cells.



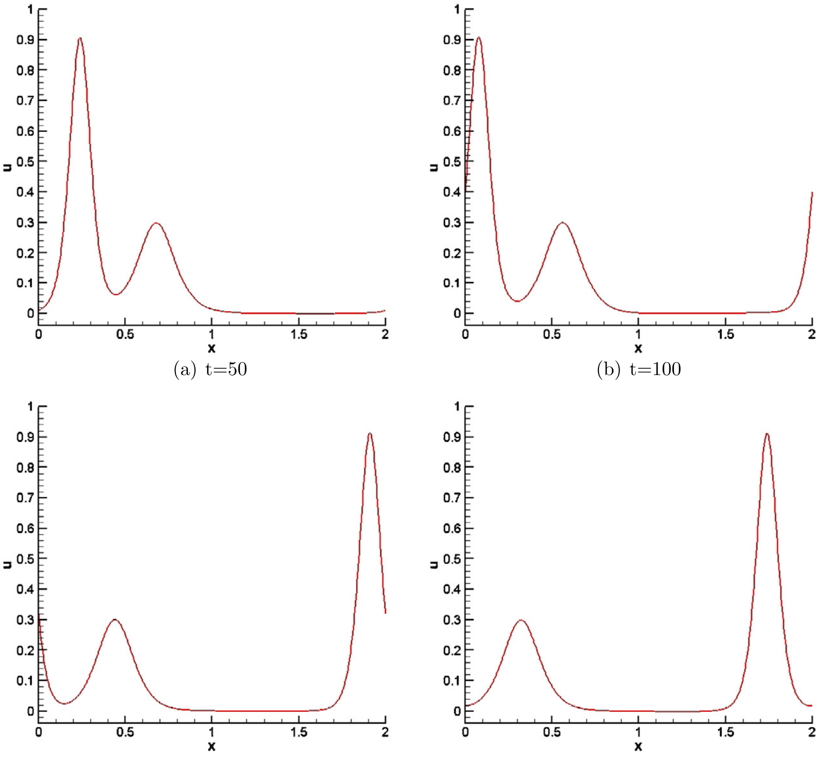
(a) t=50

(b) t=100

(c) t=150

(d) t=200

Fig. 4.4. Example 4.3: Single soliton case. P^2 elements are used with 640 cells.



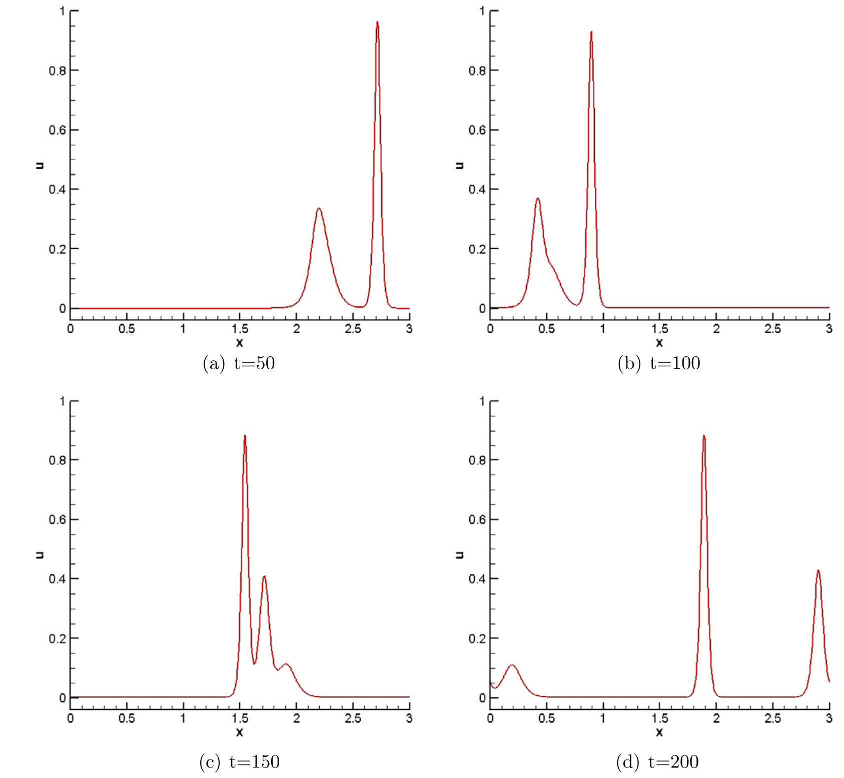
(a) t=50

(b) t=100

(c) t=150

(d) t=200

Fig. 4.5. Example 4.3: Double soliton collision case. P^2 elements are used with 640 cells.



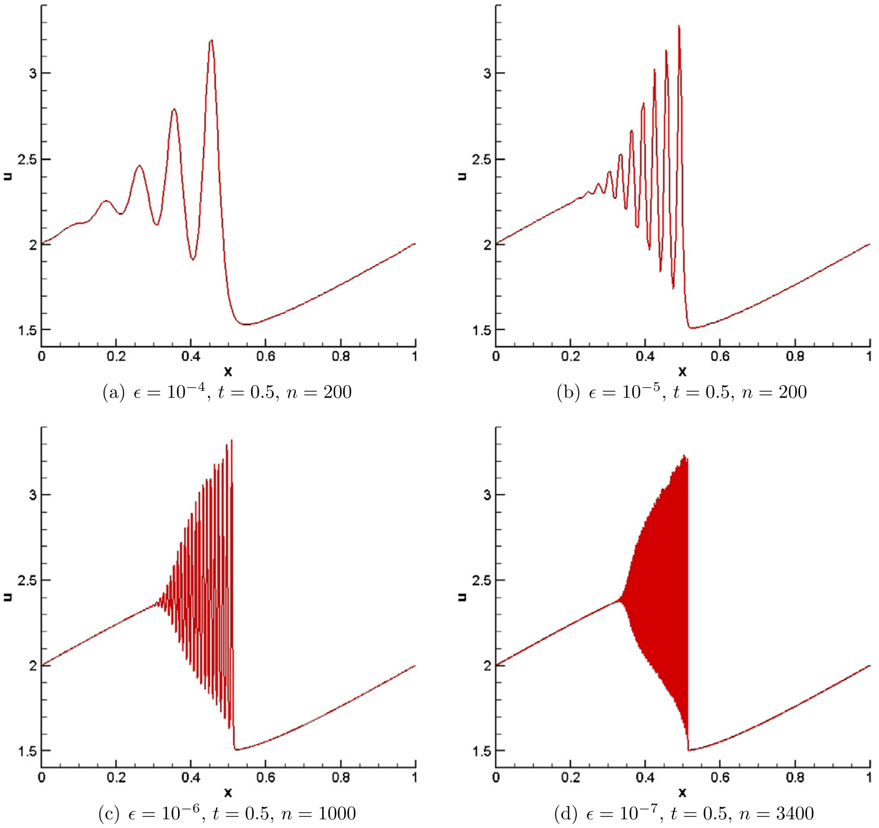
(a) t=50

(b) t=100

(c) t=150

(d) t=200

Fig. 4.6. Example 4.3: Triple soliton splitting case. P^2 elements are used with 640 cells.



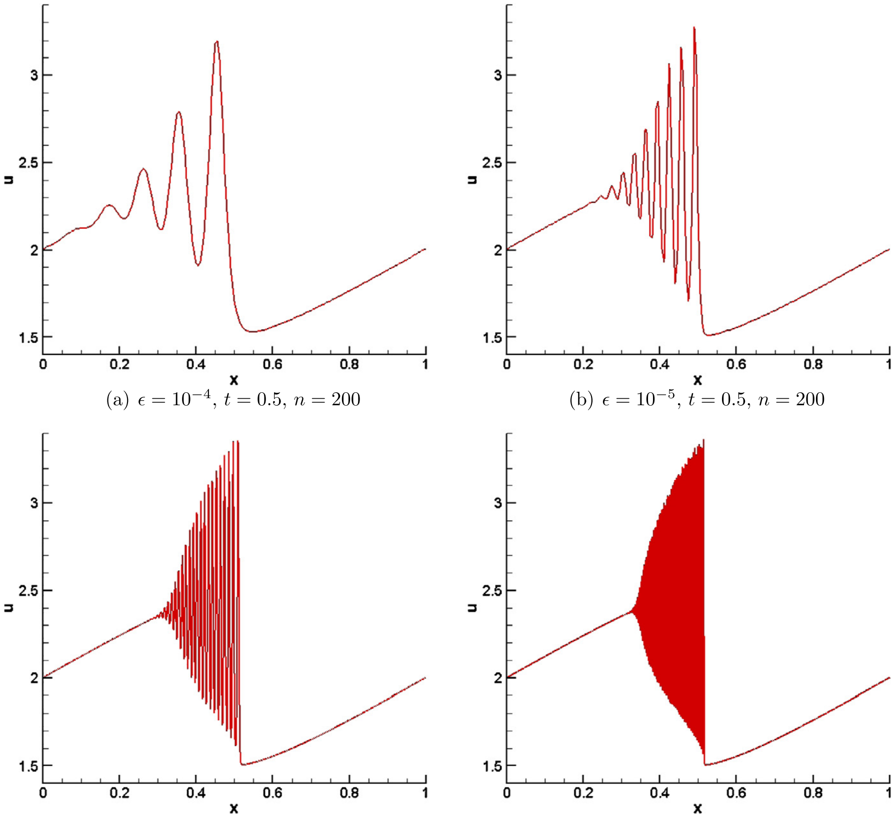
(a) $\epsilon = 10^{-4}$, $t = 0.5$, $n = 200$

(b) $\epsilon = 10^{-5}$, $t = 0.5$, $n = 200$

(c) $\epsilon = 10^{-6}$, $t = 0.5$, $n = 1000$

(d) $\epsilon = 10^{-7}$, $t = 0.5$, $n = 3400$

Fig. 4.7. Example 4.4: Zero dispersion limit of conservation laws. P^2 elements are used.



(a) $\epsilon = 10^{-4}$, $t = 0.5$, $n = 200$

(b) $\epsilon = 10^{-5}$, $t = 0.5$, $n = 200$

(c) $\epsilon = 10^{-6}$, $t = 0.5$, $n = 1000$

(d) $\epsilon = 10^{-7}$, $t = 0.5$, $n = 3400$

Fig. 4.8. Example 4.4: Zero dispersion limit of conservation laws. P^4 elements are used.

4.2. Two dimensional examples

Example 4.5. In this example we compute the solution of the linear equation

$$u_t + u_{xxx} + u_{yyy} = 0, \quad (x, y) \in (0, 2\pi) \times (0, 2\pi)$$

with the initial condition $u(x, y, 0) = \sin(x + y)$ and the periodic boundary condition in both directions. Here, $f_1(u) = f_2(u) = 0$, $g_{12} = g_{21} = 0$, $g_{11}(r) = g_{22}(r) = r$, $r_1(u) = r_2(u) = u$. The exact solution is given by $u(x, y, t) = \sin(x + y - 2t)$. The L^1, L^2, and L^∞ norm of the error and the convergence order of the scheme are shown in Table 4.3. It can be seen clearly that the scheme with P^k elements in DG gives the $(k + 1)$th order of convergence.

Example 4.6. In this example [22] we consider the Zakharov–Kuznetsov (ZK) equation

$$u_t + uu_x + \varepsilon(u_{xxx} + u_{yyx}) = 0. \tag{4.2}$$

Here, $f_1(u) = \frac{u^2}{2}$, $f_2(u) = 0$, $g_{12} = g_{22} = 0$, $g_{11}(r) = g_{21}(r) = \epsilon r$, $r_1(u) = r_2(u) = u$. The steady progressive wave solution is of the form

$$u(x, y, t) = 3\text{csech}^2(0.5\sqrt{\frac{c}{\varepsilon}}((x - ct - x_0)\cos\theta + (y - y_0)\sin\theta)), \tag{4.3}$$

where θ is an inclined angle with respect to the x-axis and (x_0, y_0) is the location of the peak of the initial u. We can see in Tables 4.4 and 4.5 that the method with P^k elements gives the $(k + 1)$th order of convergence. The computational domains are $(-16, 16) \times (-16, 16)$ in Table 4.4 and $(0, 48) \times (0, 16)$ in Table 4.5, respectively.

Example 4.7. In this example [22] we show the steady progressive wave propagation of the ZK equation (4.2). First, we show the single steady progressive wave in Figs. 4.9 and 4.10 with the initial condition

$$u(x, y, 0) = 3\text{csech}^2(0.5\sqrt{\frac{c}{\varepsilon}}((x - x_0)\cos\theta + (y - y_0)\sin\theta)), \tag{4.4}$$

Table 4.3
Example 4.5: periodic boundary conditions and $T = 1$.

k	$n \times n$	10×10	20×20	30×30	40×40	50×50
2	L^1	3.609e−3	3.576e−4	9.786e−5	3.972e−5	1.988e−5
	Order		3.34	3.20	3.13	3.10
	L^2	3.756e−3	3.788e−4	1.050e−4	4.300e−5	2.165e−5
	Order		3.31	3.16	3.10	3.08
	L_∞	4.588e−3	5.091e−4	1.436e−4	5.916e−5	3.000e−5
	Order		3.17	3.12	3.08	3.04
3	L^1	1.902e−4	7.385e−6	1.219e−6	3.538e−7	1.380e−7
	Order		4.69	4.44	4.30	4.22
	L^2	1.913e−4	7.780e−6	1.308e−6	3.830e−7	1.503e−7
	Order		4.62	4.40	4.27	4.19
	L_∞	2.418e−4	1.037e−5	1.789e−6	5.273e−7	2.083e−7
	Order		4.54	4.33	4.25	4.16

Table 4.4
Example 4.6: periodic boundary condition in both directions, $c = 0.01$, $\varepsilon = 0.01$, $\theta = 0$, $x_0 = 0$, $y_0 = 0$, $T = 1$.

k	$n \times n$	40×40	50×50	60×60	70×70	80×80
2	L^1	1.043e−6	3.151e−7	1.162e−7	5.747e−8	3.218e−8
	Order		5.36	5.47	4.88	3.98
	L^2	2.769e−6	8.216e−7	3.108e−7	1.449e−7	8.037e−8
	Order		5.44	5.33	4.95	4.41
	L_∞	1.225e−5	3.639e−6	1.740e−6	7.808e−7	4.106e−7
	Order		5.44	4.05	5.20	4.81
3	L^1	4.540e−7	9.451e−8	2.918e−8	1.043e−8	4.314e−9
	Order		7.03	6.45	6.67	6.61
	L^2	1.200e−6	2.658e−7	7.797e−8	2.757e−8	1.125e−8
	Order		6.76	6.73	6.74	6.71
	L_∞	5.612e−6	1.210e−6	3.307e−7	1.110e−7	4.257e−8
	Order		6.88	7.11	7.08	7.18

Table 4.5
Example 4.6: periodic boundary condition in the x-direction and Dirichlet boundary condition in the y-directions, $c = 0.01$, $\varepsilon = 0.01$, $\theta = \frac{\pi}{12}$, $x_0 = 24$, $y_0 = 8$, and $T = 1$.

k	$n \times n$	60×60	70×70	80×80	90×90	100×100
2	L^1	5.366e−7	2.352e−7	1.133e−7	6.017e−8	3.562e−8
	Order		5.35	5.47	5.37	4.98
	L^2	1.821e−6	7.759e−7	3.709e−7	1.975e−7	1.159e−7
	Order		5.53	5.53	5.35	5.06
	L_∞	1.222e−5	5.137e−6	2.753e−6	1.710e−6	9.783e−7
	Order		5.62	4.67	4.04	5.30
3	L^1	2.561e−7	8.737e−8	3.563e−8	1.644e−8	8.264e−9
	Order		6.98	6.72	6.57	6.53
	L^2	8.420e−7	2.931e−7	1.205e−7	5.491e−8	2.718e−8
	Order		6.85	6.66	6.67	6.67
	L_∞	5.145e−6	1.977e−6	9.909e−7	3.976e−7	2.441e−7
	Order		6.20	5.17	7.75	4.63

where θ is the inclined angle with respect to the x-axis. The periodic boundary condition in both x- and y-directions are used when $\theta = 0$. Since the solution cannot be periodic in y-direction when $\theta \neq 0$, we use for this case the Dirichlet boundary condition in the y-direction and a periodic boundary condition in the x-direction.

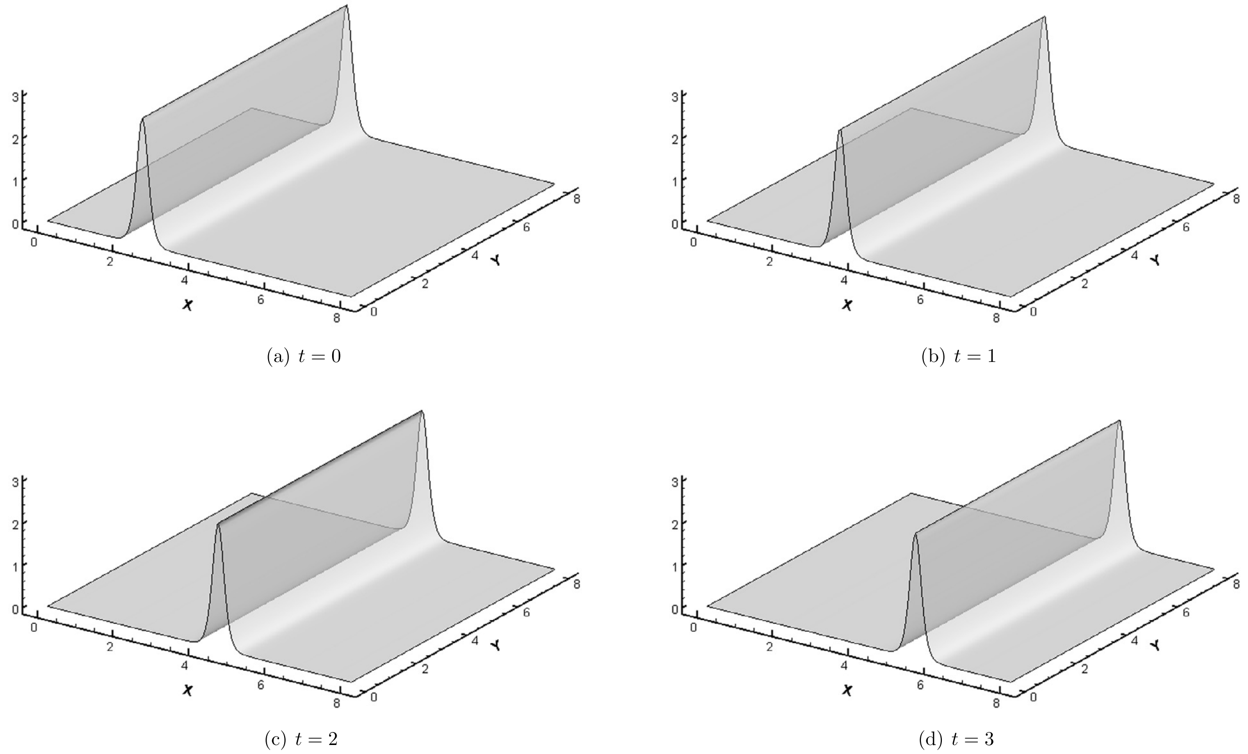
The double soliton collision case has the initial condition

$$u(x, y, 0) = \sum_{j=1}^{2} 3c_j \mathrm{sech}^2(0.5\sqrt{\frac{c_j}{\varepsilon}}((x - x_j)\cos\theta + (y - y_j)\sin\theta)), \tag{4.5}$$

where $c_1 = 0.45$, $c_2 = 0.25$, $\varepsilon = 0.01$, $\theta = 0$, $x_1 = 2.5$, $y_1 = 0$, $x_2 = 3.3$, $y_2 = 0$. The results with periodic boundary conditions in both coordinate directions in $(0, 8) \times (0, 8)$ using P^2 elements with 150×150 uniform cells are shown in Fig. 4.11.

Example 4.8. In this example [22] we show the numerical results for the equation

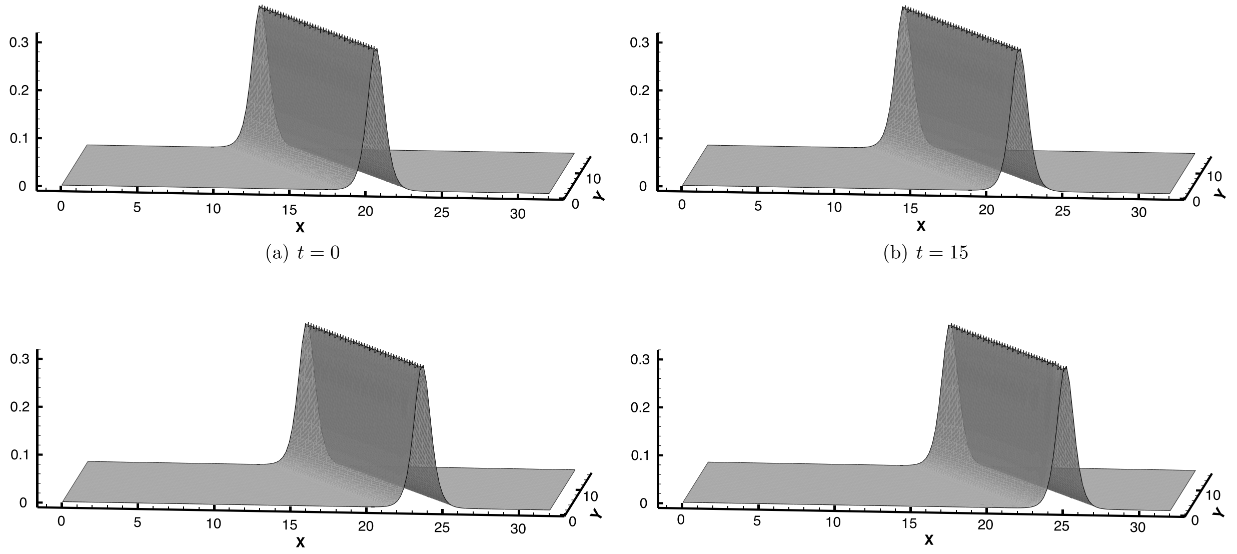
$$u_t + (3u^2)_x + u_{xxx} + u_{xyy} = 0. \tag{4.6}$$



(a) $t=0$ (b) $t=1$

(c) $t=2$ (d) $t=3$

Fig. 4.9. Example 4.7: The single steady progressive wave propagation with the initial condition (4.4). $c=1$, $\varepsilon=0.01$, $\theta=0$, $x_0=2.5$, $y_0=4$. Periodic boundary condition in both coordinate directions in $(0,8)\times(0,8)$. P^2 elements with 150×150 uniform cells are used.

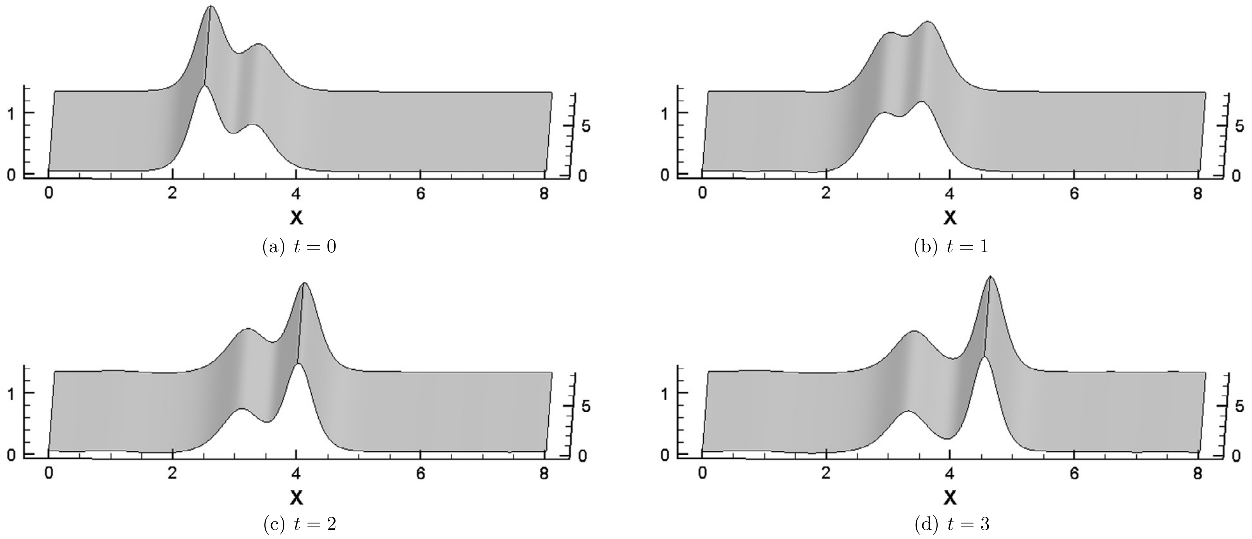


(a) $t=0$ (b) $t=15$

(c) $t=30$ (d) $t=45$

Fig. 4.10. Example 4.7: The single steady progressive wave propagation with the initial condition (4.4). $c=0.1$, $\varepsilon=0.01$, $\theta=\frac{\pi}{6}$, $x_0=16$, $y_0=8$. Periodic boundary condition in the x-direction and Dirichlet boundary condition in the y-direction in $(0,32)\times(0,16)$. P^2 elements with 150×150 uniform cells are used.

Here, $f_1(u)=3u^2$, $f_2(u)=0$, $g_{12}=g_{22}=0$, $g_{11}(u)=g_{21}(u)=u$, $r_1(u)=r_2(u)=u$. A cylindrically symmetric solitary solution was obtained and its evolutions as well as interactions were investigated numerically in [10]. This type of solitary solution, also called the bell-shaped pulse, can be expressed as



(a) $t = 0$

(b) $t = 1$

(c) $t = 2$

(d) $t = 3$

Fig. 4.11. Example 4.7: Double soliton collision profiles and periodic boundary condition in both coordinate directions in $(0, 8) \times (0, 8)$. P^2 elements with 150×150 uniform cells are used.

Table 4.6
Example 4.8: Coefficients for the solitary wave solution of the equation (4.6).

n	a_{2n}	n	a_{2n}
1	−1.25529873	6	−0.00281281
2	0.21722635	7	−0.00138352
3	0.06452543	8	−0.00070289
4	0.00540862	9	−0.00020451
5	−0.00332515	10	−0.00003053

$$u(x, y, t) = \frac{c}{3}\sum_{n=1}^{10} a_{2n}(\cos(2n \operatorname{arccot}(\frac{\sqrt{c}}{2}r)) - 1), \tag{4.7}$$

where c is the velocity of the solitary wave solution and $r = \sqrt{(x - ct)^2 + y^2}$. The coefficients are collected in Table 4.6. We use the Dirichlet boundary condition given by the exact solution. The stable propagation of a single pulse is shown in Fig. 4.12.

Then, we proceed to show the collision of two pulses with the initial condition

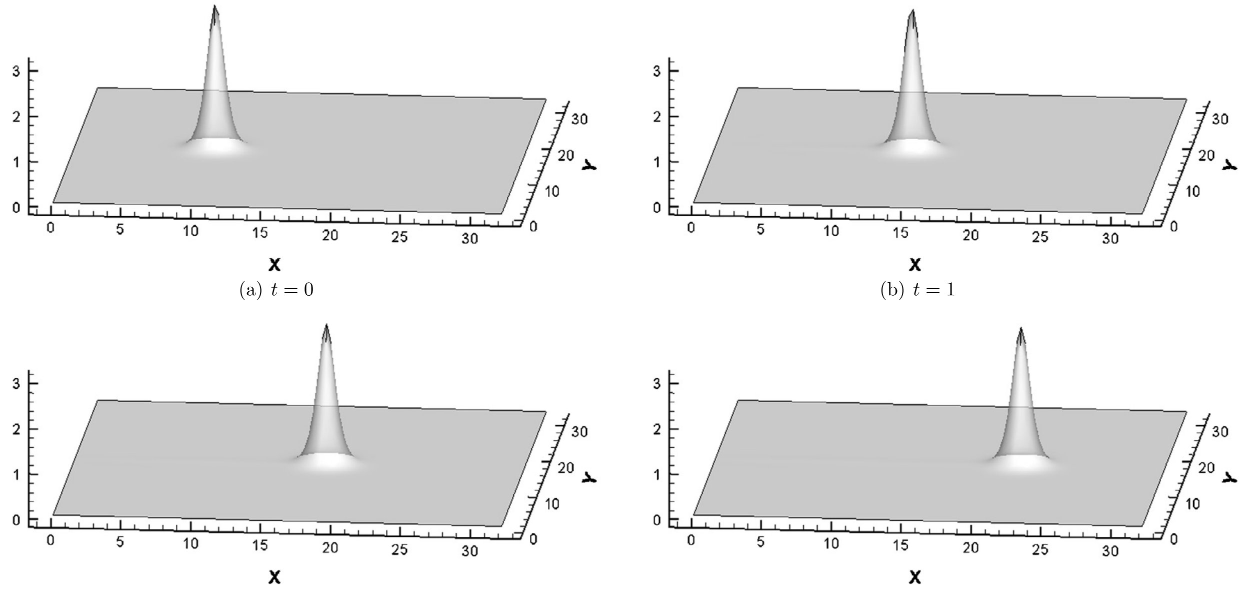
$$u(x, y, 0) = \sum_{j=1}^{2} \frac{c_j}{3}\sum_{n=1}^{10} a_{2n}(\cos(2n \operatorname{arccot}(\frac{\sqrt{c_j}}{2}r_j)) - 1), \tag{4.8}$$

where c_1 and c_2 are the velocities of the solitary wave solutions, $r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$ $(i = 1, 2)$, and (x_i, y_i)'s are the locations of the peaks of u. When the centers of the two pulses are situated on the same line with $y = \text{const}$, the collision is called a direct collision and otherwise called a deviated collision [10]. The numerical solutions obtained for the cases of a direct collision and a deviated collision of two pulses are shown in Figs. 4.13 and 4.14, respectively.

5. Conclusions and further comments

In the previous sections we have studied a hybrid LDG-HWENO scheme for solving KdV-type equations. The scheme uses the cell averages of the physical solution and first moment(s) as unknown variables (a feature of HWENO) while approximates high order spatial derivatives using the local DG method. It has less unknown variables than a pure LDG method and can be applied to problems involving high order spatial derivatives. Numerical results have been presented for a selection of one and two dimensional linear and nonlinear examples. They confirm the designed convergence order of the scheme.

The new scheme employs an explicit TVD Runge–Kutta method for time integration and is subject to the CFL condition which requires an extremely small time step restriction of the form $\Delta t = \mathcal{O}(\Delta x^3)$. How to avoid this restriction using suitable implicit time discretization will be an interesting topic for future work. In this aspect, it is worth pointing out



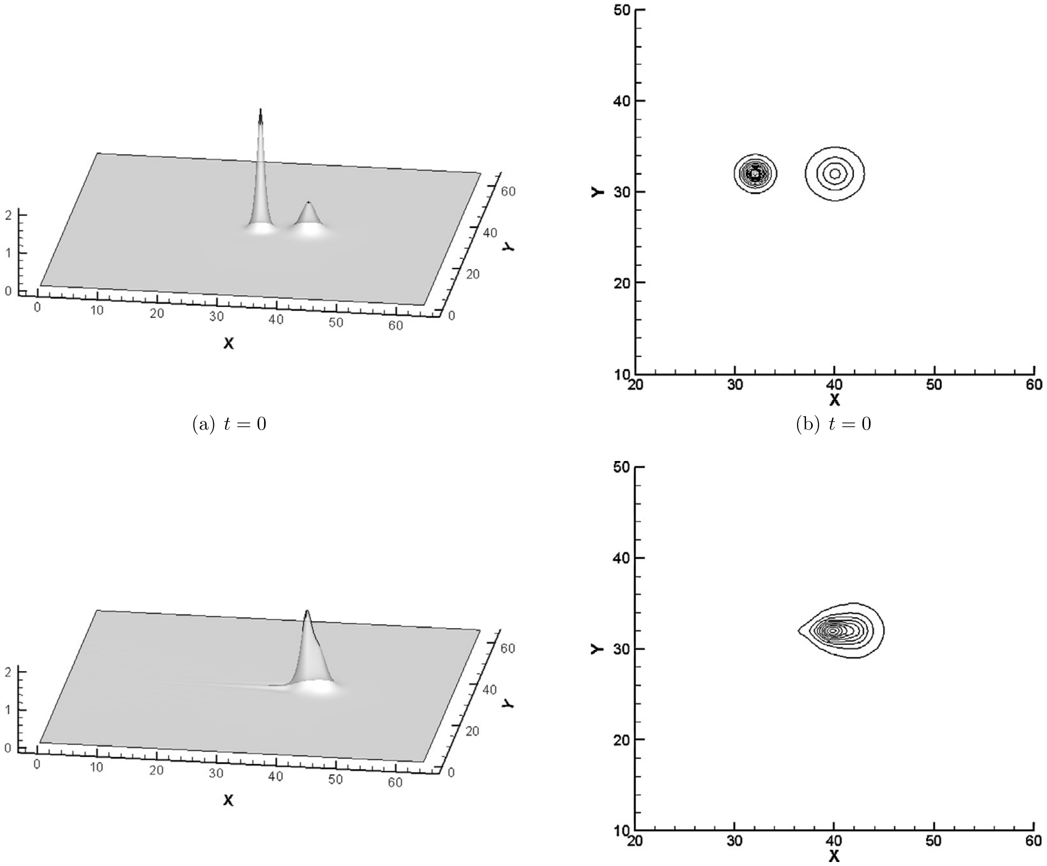
(a) $t = 0$

(b) $t = 1$

(c) $t = 2$

(d) $t = 3$

Fig. 4.12. Example 4.8: Evolution of a single bell-shaped pulse solution for (4.6) with $c = 4$, $x_0 = 10$, and $y_0 = 16$. The used domain is $(0, 32) \times (0, 32)$. P^2 elements with 100×100 uniform cells are used.



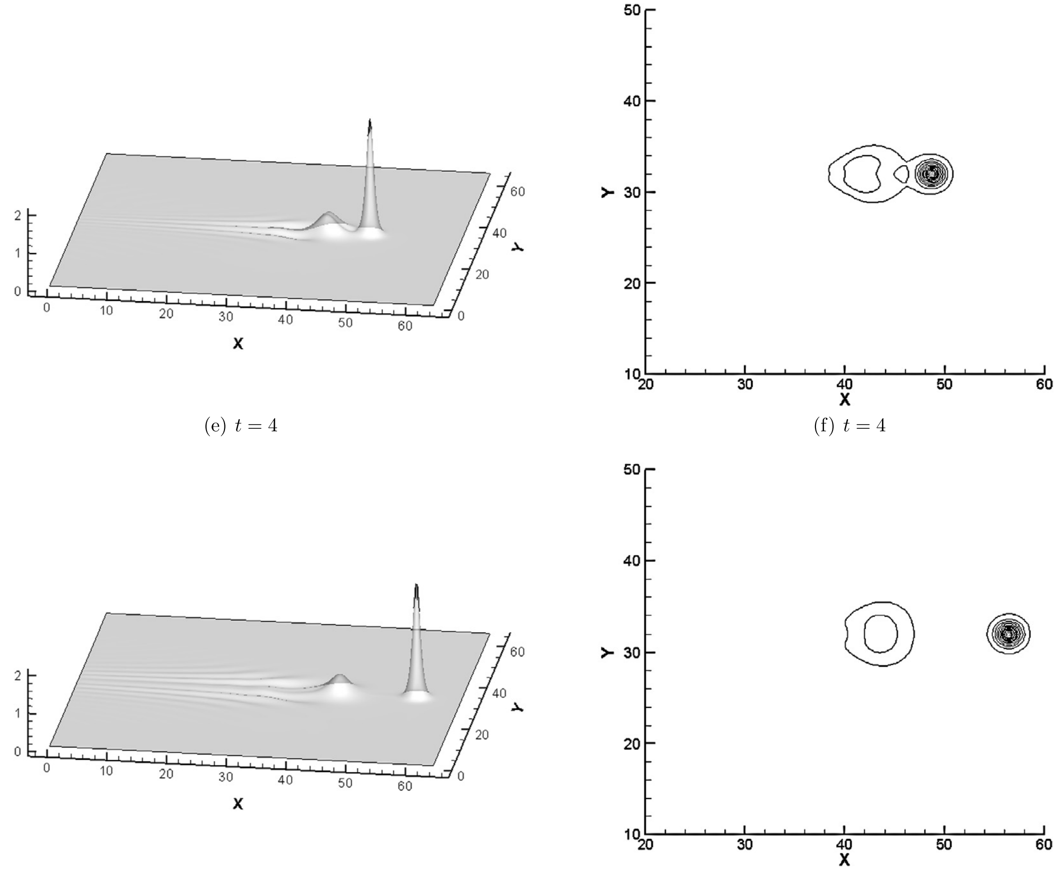
(a) $t = 0$

(b) $t = 0$

(c) $t = 2$

(d) $t = 2$

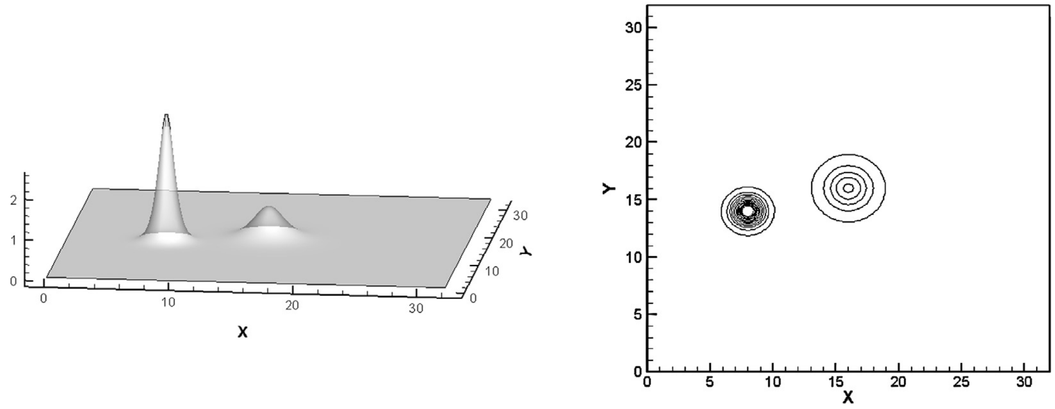
Fig. 4.13. Example 4.8: Direct collision of two bell-shaped pulses solution for (4.6) with the initial condition (4.8) and $c_1 = 4$, $c_2 = 1$, $x_1 = 32$, $y_1 = 32$, $x_2 = 40$, $y_2 = 32$, and Dirichlet boundary condition. The computational domain is $(0, 64) \times (0, 64)$. P^2 elements with 200×200 uniform cells are used.



(e) $t = 4$ (f) $t = 4$

(g) $t = 6$ (h) $t = 6$

Fig. 4.13. (*continued*)

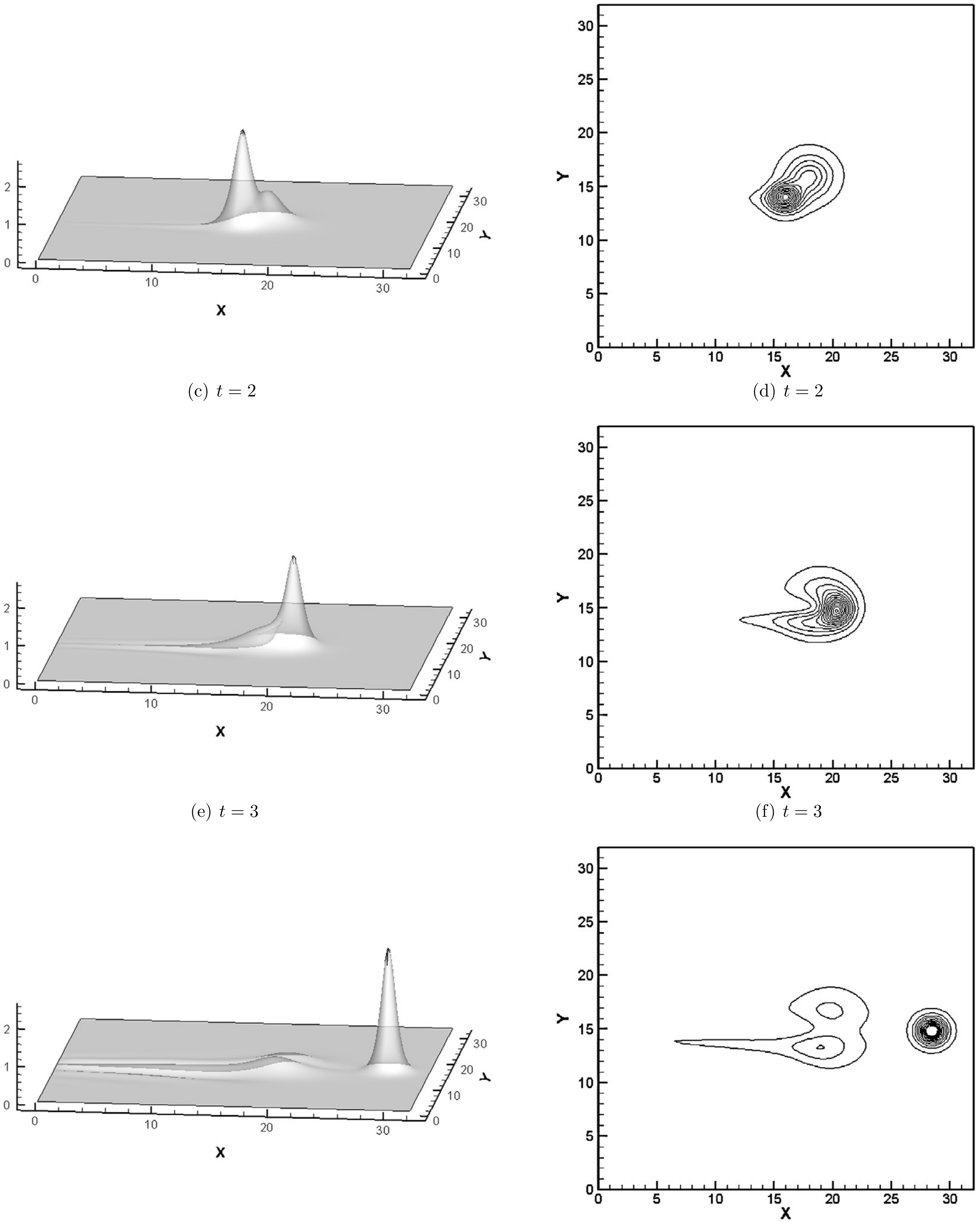


(a) $t = 0$ (b) $t = 0$

Fig. 4.14. Example 4.8: Deviated collision of two bell-shaped pulses solution for (4.6) with the initial condition (4.8) and $c_1 = 4$, $c_2 = 1$, $x_1 = 8$, $y_1 = 14$, $x_2 = 16$, $y_2 = 16$. and Dirichlet boundary condition. The computational domain is $(0, 32) \times (0, 32)$. P^2 elements with 150×150 uniform cells are used.

that an unconditionally stable LDG scheme has been recently proposed by Dumbser and Facchini [9] for Boussinesq-type equations. The scheme, based on space–time discontinuous Galerkin finite elements, avoids the severe restriction on the time step.

Finally, we point out that the scheme proposed in this work can be extended to unstructured meshes with slight modifications. In that case, the dimension-by-dimension HWENO reconstruction used in this work, which can be used only for rectangular meshes, should be replaced with the genuine 2D version of HWENO reconstruction [17] that works for unstructured meshes.



(c) $t = 2$

(d) $t = 2$

(e) $t = 3$

(f) $t = 3$

(g) $t = 5$

(h) $t = 5$

Fig. 4.14. (*continued*)

References

[1] D.S. Balsara, C. Altmann, C.-D. Munz, M. Dumbser, A sub-cell based indicator for troubled zones in RKDG schemes and a novel class of hybrid RKDG+HWENO schemes, J. Comput. Phys. 226 (2007) 586–620.

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

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

[4] B. Cockburn, S. Hou, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws IV: the multidimensional case, Math. Comput. 54 (1990) 545–581.

[5] B. Cockburn, 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) 199–224.

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

[7] M. Dumbser, Arbitrary high order PNPM schemes on unstructured meshes for the compressible Navier–Stokes equations, Comput. Fluids 39 (2010) 60–76.

[8] M. Dumbser, D.S. Balsara, E.F. Toro, C.D. Munz, A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes, J. Comput. Phys. 227 (2008) 8209–8253.

[9] M. Dumbser, M. Facchini, A local space–time discontinuous Galerkin method for Boussinesq-type equations, Appl. Math. Comput. 272 (2016) 336–346.

[10] H. Iwasaki, S. Toh, T. Kawahara, Cylindrical quasi-solitons of the Zakharov–Kuznetsov equation, Physica D 43 (1990) 293–303.

[11] G. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.

[12] X. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.

[13] H. Luo, L. Luo, R. Nourgaliev, V.A. Mousseau, N. Dinh, A reconstructed discontinuous Galerkin method for the compressible Navier–Stokes equations on arbitrary grids, J. Comput. Phys. 229 (2010) 6961–6978.

[14] J. Qiu, C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method: one-dimensional case, J. Comput. Phys. 193 (2003) 115–135.

[15] J. Qiu, C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method II: two dimensional case, Comput. Fluids 34 (2005) 642–663.

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

[17] J. Shi, C. Hu, C.-W. Shu, A technique of treating negative weights in WENO schemes, J. Comput. Phys. 175 (2002) 108–127.

[18] C.-W. Shu, Total-variation-diminishing time discretizations, SIAM J. Sci. Stat. Comput. 9 (1988) 1073–1084.

[19] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.

[20] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, II, J. Comput. Phys. 83 (1989) 32–78.

[21] Y. Xu, C.-W. Shu, Local discontinuous Galerkin methods for high-order time-dependent partial differential equations, Commun. Comput. Phys. 7 (2010) 1–46.

[22] Y. Xu, C.-W. Shu, Local discontinuous Galerkin methods for two classes of two-dimensional nonlinear wave equations, Physica D 208 (2005) 21–58.

[23] J. Yan, C.-W. Shu, A local discontinuous Galerkin method for KdV type equations, SIAM J. Numer. Anal. 40 (2002) 769–791.

[24] F. Zheng, J. Qiu, Directly solving the Hamilton–Jacobi equations by Hermite WENO Schemes, J. Comput. Phys. 307 (2016) 423–445.