Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 227 (2007) 376–399

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Numerical method satisfying the first two conservation laws for the Korteweg–de Vries equation ☆

Cui Yanfen, Mao De-kang *

Department of Mathematics, Shanghai University, No. 99, ShangDa Rd., Shanghai 200444, PR China

Received 25 April 2007; received in revised form 27 July 2007; accepted 28 July 2007
Available online 11 August 2007

Abstract

In this paper, we develop a finite-volume scheme for the KdV equation which conserves both the momentum and energy. The main ingredient of the method is a numerical device we developed in recent years that enables us to construct numerical method for a PDE that also simulates its related equations. In the method, numerical approximations to both the momentum and energy are conservatively computed. The operator splitting approach is adopted in constructing the method in which the conservation and dispersion parts of the equation are alternatively solved; our numerical device is applied in solving the conservation part of the equation. The feasibility and stability of the method is discussed, which involves an important property of the method, the so-called Jensen condition. The truncation error of the method is analyzed, which shows that the method is second-order accurate. Finally, several numerical examples, including the Zabusky–Kruskal's example, are presented to show the good stability property of the method for long-time numerical integration.

AMS subject classifications: 65M06; 35Q53

Keywords: KdV equation; Conservation of momentum and energy; Jensen condition; Truncation error

1. Introduction

It is well known that the Korteweg–de Vries (KdV) equation

$$u_t + \left(\frac{1}{2}u^2\right)_x + \varepsilon u_{xxx} = 0, \tag{1.1}$$

where ε is a constant, possesses an infinite set of conservation laws and the first two of them are Eq. (1.1), which describes the conservation of momentum and

$$(u^2)_t + \left(\frac{2}{3}u^3\right)_x + \varepsilon(2uu_{xx} - (u_x)^2)_x = 0, \tag{1.2}$$

☆ Research is supported by Shanghai Pu Jiang Program [2006] 118.

* Corresponding author. Tel.: +86 21 66134464; fax: +86 21 66133239.
E-mail address: dkmao@staff.shu.edu.cn (D.-k. Mao).

doi:10.1016/j.jcp.2007.07.031

which describes the conservation of energy. It is commonly believed that it is good for a numerical method for the KdV equation to simulate as many of these conservation relations as possible; methods preserving more conservation relations are usually more stable and suitable for long-time integration. However, maintaining more than one conservation relations is difficult in the practice of numerical simulation. From a broader point of view of mathematics, the difficulty is that there is generally no enough degrees of freedom to construct numerical methods for a PDE that also simulates the equations that are derived from the original one.

A lot of numerical methods have been developed for the KdV equation in recent years, ranging from finite difference methods, finite element methods to spectral methods, see [25,8,16,24,23,3,1,2,20,26,5,6] and the references cited therein. Most of them preserve the conservation of momentum and only a few of them preserve the conservation of both the momentum and energy. We should particularly mention the symplectic and multisymplectic schemes, see [3,1,2,26,20]. When viewing the KdV equation as a Hamiltonian system, these schemes preserve the symplectic structure of the system. Therefore, they are proved to be good numerical methods for long-time integration of the equation. However, symplectic and multisymplectic schemes usually do not exactly conserve the momentum and energy and thus how closely the momentum and energy are conserved in computation is then a property of interest for these schemes and is often assessed in numerical experiments, see the references cited above.

In this paper, we develop a finite difference method for the KdV equation which satisfies both the momentum and energy conservation relations. The main ingredient of the method is a device we developed in recent years that enables us to construct numerical methods that simultaneously simulates a PDE and its related equations, see [11,4,12–14,21,22]. To explain the idea, we look at the linear advection equation

$$u_t + u_x = 0. \tag{1.3}$$

This equation possesses infinitively many conservation laws. As a matter of fact, for any smooth function $U(u)$ the following equation

$$U(u)_t + U(u)_x = 0 \tag{1.4}$$

is also satisfied. Both u and $U(u)$ are then conserved in the sense that

$$\int_{-\infty}^{\infty} u(x,t)\mathrm{d}x = \int_{-\infty}^{\infty} u(x,0)\mathrm{d}x \tag{1.5}$$

and

$$\int_{-\infty}^{\infty} U(u(x,t))\mathrm{d}x = \int_{-\infty}^{\infty} U(u(x,0))\mathrm{d}x. \tag{1.6}$$

In the following, we are going to construct a conservative numerical scheme that simulates both Eqs. (1.3) and (1.4) for a given $U(u)$.

Our scheme is of the Godunov type and its numerical solution $\{u_j^n\}$ is a cell-average approximation to the exact solution at time t_n

$$u_j^n \simeq \frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x,t_n)\mathrm{d}x. \tag{1.7}$$

A very special feature of the scheme is that it computes also a cell-average approximation to $U(u)$

$$U_j^n \simeq \frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} U(u(x,t_n))\mathrm{d}x. \tag{1.8}$$

Like all the Godunov-type schemes (see [9,10]), the scheme proceeds in the *reconstruction*, *evolution* and *cell-averaging* steps.

In the reconstruction step, the solution is reconstructed in each cell $(x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}})$ as a linear function

$$R(x;u^n,U^n) = u_j^n + s_j^n(x - x_j), \quad j = \ldots,-1,0,1,\ldots, \tag{1.9}$$

where s_j^n is the slope. Another very special feature of the scheme is that the slope is not computed by interpolating the solution as is done in ordinary finite-volume schemes, but rather by requiring

$$\frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} U(R(x;u^n,U^n))\mathrm{d}x = U_j^n, \tag{1.10}$$

that is, the cell-average of $U(u)$ of the reconstructed solution must be equal to the numerical cell-average of $U(u)$ in the grid cell. Eq. (1.10) is an equation of s_j^n, from which we solve out the slope.

The evolution step is to solve the IVP

$$\begin{cases} v_t + v_x = 0, \\ v(x,t_n) = R(x;u^n,U^n), \end{cases} \quad t_n < t < t_{n+1}, \tag{1.11}$$

as that in all the Godunov's type schemes do and the cell-averaging step is to cell-average v and $U(v)$ to Eq. (1.11) at $t = t_{n+1}$ to obtain u_j^{n+1} and U_j^{n+1}. In practice, we use the integral form of Eqs. (1.3) and (1.4) to compute u_j^{n+1} and U_j^{n+1}, which results in the following conservative schemes

$$u_j^{n+1} = u_j^n - \lambda(\hat{f}_{j+1/2}^n - \hat{f}_{j-1/2}^n) \tag{1.12}$$

and

$$U_j^{n+1} = U_j^n - \lambda(\widehat{F}_{j+1/2}^n - \widehat{F}_{j-1/2}^n), \tag{1.13}$$

where $\hat{f}_{j+1/2}^n$ and $\widehat{F}_{j+1/2}^n$ are the flux approximations to u and $U(u)$ in Eqs. (1.3) and (1.4), respectively, on the cell boundaries.

The scheme so constructed maintains the conservation for both u and $U(u)$. We note that our numerical approximations u_j^n and U_j^n are not related as customarily $U(u_j^n) = U_j^n$, but rather in a loosen fashion as in Eqs. (1.8) and (1.10). Actually, the numerical solution in each grid cell can be understood from Eqs. (1.8) and (1.10) as a piece of linear function whose cell-average is u_j^n and whose cell-average of $U(u)$ is U_j^n. In this fashion, we gain one degree of freedom in describing the numerical solution, which enables us to maintain the conservation relations for both u and $U(u)$ in constructing the scheme.

Along this line, schemes maintaining more conservation relations can also be constructed. This can be accomplished by involving more entities $(U_1)^n, (U_2)^n, \ldots$ in the numerical scheme, where

$$(U_i)_j^n \simeq \frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} U_i(u(x,t_n))\mathrm{d}x, \tag{1.14}$$

with $U_i(u)$ being nonlinear functions of u, either convex or not, reconstructing the solution in each cell as polynomial

$$R(x;u^n,(U_1)^n,(U_2)^n,\ldots) = u_j^n + s_1(x-x_j) + s_2(x-x_j)^2 + \cdots, \tag{1.15}$$

and solving the coefficients s_1, $s_2, \ldots$ from the system of equations

$$\frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} U_i(R(x;u^n,(U_1)^n,(U_2)^n,\ldots))\mathrm{d}x = (U_i)_j^n, \quad i = 1,2,\ldots, \tag{1.16}$$

see [21]. For more general evolution PDE's, the functions U_i's may also involve the derivatives of u, i.e. u_x,u_{xx}, ... and U_i's are not necessary to be conserved.

For continuous solutions to Eq. (1.3), the numerical results computed by this kind of schemes are fantastic. In Fig. 1, we present two numerical results computed by this kind of schemes with 200 grid cells, which are cited from [21]. Both are of the Wavepacket problem, see [9]. The result on the left is obtained by a scheme maintaining u and u^2 conserved (second-order accurate) and is at $t = 200$ and the result on the right is by a scheme maintaining u, u^2 and u^3 conserved (third-order accurate) and is at $t = 20{,}000$. As to our knowledge, no scheme up to date has ever got qualified numerical result for this example with the same grid beyond the time $t = 20$.

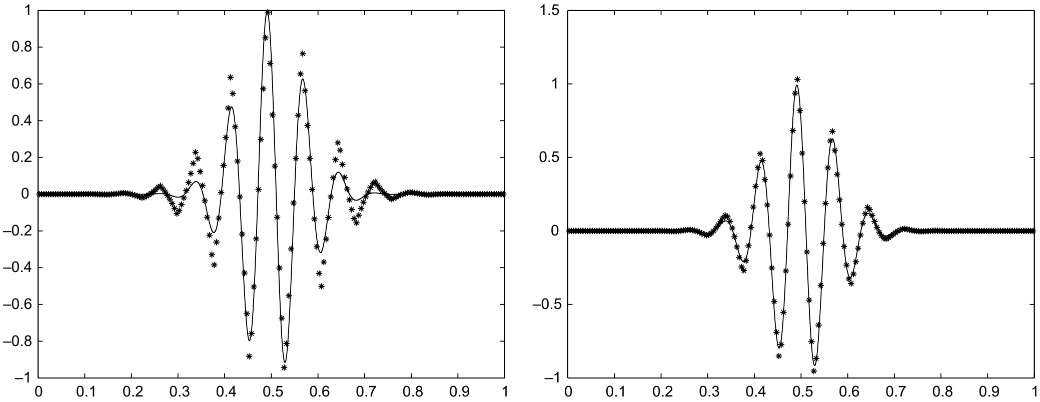


Fig. 1. The numerical result on the left is obtained by a scheme maintaining two conservation relations (second-order accurate) and is at $t = 200$, and the one on the right is obtained by a scheme maintaining three conservation relations (third-order accurate) and is at $t = 20{,}000$.

The success of our numerical device in the linear advection equation encourages us to apply it to the KdV equation since the latter also possesses many conservation laws and maintaining them in numerical simulation is important. We develop a scheme which computes and conserves both the momentum u and energy $U(u)$. That is, the constructed scheme involves two numerical entities, the numerical momentum u_j^n and numerical energy U_j^n and both of them are conservatively computed.

To construct the scheme, we adopt the splitting strategy as in [8], i.e. split the Eq. (1.1) into the conservation part (2.5) and dispersion part (2.6). Our numerical device is applied in solving the conservation part, where the solution is reconstructed in each grid cell with its momentum cell-average being u_j^n and its energy cell-average being U_j^n. To solve the dispersion part, we adopt the implicit difference scheme in [8] to compute the momentum cell-averages $\{u_j^n\}$ and once they are computed the energy cell-averages $\{U_j^n\}$ are then computed passively from them. The advantage of using this implicit scheme is that the time step τ does not suffer from a prohibiting restriction $\tau = \mathrm{O}(h^3)$, and the corresponding linear algebra system is still easy to be solved using well developed numerical methods. Therefore, the only restriction on the time step in our method comes from the conservation part, which is of $\tau = \mathrm{O}(h)$.

Constructed in such a way, our method is stable in the sense that its energy is nonnegative and L^1 bounded and its momentum is L^2 bounded by the L^1 norm of its energy, see Theorem 3.2. The method is second-order accurate away from extremes of solution and is at least first-order accurate near extremes, see Theorem 4.3.

We know that the KdV equation is notorious for its "marginal" stability that is resulted in the balance between the nonlinear convection and the linear dispersion, see [2]. This "living at the edge of stability" becomes more fragile in the "convection-dominated" cases, in which the linear dispersion is very weak compared to the nonlinear convection; therefore, long-time numerical integrations for these cases are difficult since they tend to become unstable. A typical case is the so-called Zabusky–Kruskal's example, see [25], on which many numerical schemes, including the Zabusky–Kruskal's scheme, explode in long-time numerical integrations, see [1,18,20,26,5]. We test our method on "convection-dominated" cases, especilally the Zabusky–Kruskal's example and the numerical results are fantastic. The method is very stable and robust in long-time integrations and the solitions are very smooth with well-preserved shapes of solitons. The numerical simulation for the Zabusky–Kruskal's example goes up to the time $t = 20t_{\mathrm{R}}$ and still does not exhibit any blowup. It seems to us that the computation can go on for ever.

The organization of the paper is as follows: Section 1 is the introduction. In Section 2, we describe the numerical method in detail. Section 3 discusses the feasibility and stability of the method, which involves a very important property of the method, the so-called Jensen condition. Section 4 discusses the accuracy of the method and proves that the method is essentially second-order accurate. In Section 5, we present several numerical examples computed with our method, which include the Zabusky–Kruskal's example [25], to show its good stability property for long-time numerical integration. Finally, Section 6 is the conclusion.

2. Description of the method

2.1. Numerical solution

We consider the Cauchy problem for the KdV equation

$$\begin{aligned} & u_t + \left(\frac{1}{2}u^2\right)_x + \varepsilon u_{xxx} = 0, \\ & u(x,0) = u_0(x), \end{aligned} \tag{2.1}$$

with a suitable initial function $u_0(x)$, where ε is a constant which may be very small. As we have mentioned in the previous section, Eq. (2.1) possesses infinitely many conservation laws and the first two of them are Eq. (2.1) and

$$(u^2)_t + \left(\frac{2}{3}u^3\right)_x + \varepsilon(2uu_{xx} - (u_x)^2)_x = 0. \tag{2.2}$$

Eq. (2.1) is the conservation law for the momentum u and Eq. (2.2) is the conservation law for the energy $U(u) = u^2$.

For numerical discretization, we consider only uniform grids and use the notations $x_j = jh,\ x_{j\pm\frac{1}{2}} = \left(j \pm \frac{1}{2}\right)h$ and $t_n = n\tau$, with h and τ being the spatial and temporal increments, respectively. Our method is of the finite-volume type (see [9,10]) whose solution is a cell-average approximation to the true solution

$$u_j^n \simeq \frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x,t_n)\mathrm{d}x. \tag{2.3}$$

A very special feature of our method is that it also computes a cell-average approximation to the energy $U(u)$ of the true solution,

$$U_j^n \simeq \frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} U(u(x,t_n))\mathrm{d}x = \frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u^2(x,t_n)\mathrm{d}x. \tag{2.4}$$

In this way, the solution to Eq. (2.1) is understood as consisting of two entities, the momentum u and the energy $U(u) = u^2$ and they are evolved with time by Eqs. (2.1) and (2.2), respectively. Likewise, the numerical solution also consists of approximations to the two entities, numerical momentum u_j^n and numerical energy U_j^n, in each grid cell.

2.2. Operator splitting

As in [8], we apply the method of operator splitting to solve Eq. (2.1); i.e. we are going to alternately solve the conservation part and dispersion part of Eq. (2.1). However, with the above understanding of solution to Eq. (2.1), the conservation and dispersion parts are defined, respectively, as

$$\begin{aligned} & u_t + \left(\frac{1}{2}u^2\right)_x = 0, \\ & (u^2)_t + \left(\frac{2}{3}u^3\right)_x = 0, \\ & u(x,0) = u_0(x), \quad (u(x,0))^2 = (u_0(x))^2, \end{aligned} \tag{2.5}$$

and

$$\begin{aligned} & u_t + \varepsilon u_{xxx} = 0, \\ & (u^2)_t + \varepsilon(2uu_{xx} - (u_x)^2)_x = 0, \\ & u(x,0) = u_0(x), (u(x,0))^2 = (u_0(x))^2. \end{aligned} \tag{2.6}$$

Let A_t be the solution operator associated with the conservation part (2.5); i.e. we write the unique momentum and energy to Eq. (2.5) as

$$\begin{bmatrix} u(x,t) \\ U(u(x,t)) \end{bmatrix} = A_t \begin{bmatrix} u_0(x) \\ U(u_0(x)) \end{bmatrix}. \tag{2.7}$$

Similarly, we denote the solution operator associated with the linear dispersion part by B_t. Then we approximate the solution to Eq. (2.1) by the Godunov's splitting

$$\begin{bmatrix} u_\tau(x,n\tau) \\ U_\tau(u(x,n\tau)) \end{bmatrix} = [B_\tau \circ A_\tau]^n \begin{bmatrix} u_0(x) \\ U(u_0(x)) \end{bmatrix} \tag{2.8}$$

for small time step τ. Of course, when this approach is implemented, both A_τ and B_τ must be replace by numerical methods.

The above Godunov's splitting method is in general only first-order accurate formally. In order to obtain second-order accuracy for the method, we may use Strang's operator splitting instead. Then we approximate the solution over time by

$$\begin{bmatrix} u_\tau(x,n\tau) \\ U_\tau(u(x,n\tau)) \end{bmatrix} = [A_{\frac{\tau}{2}} \circ B_{\frac{\tau}{2}} \circ B_{\frac{\tau}{2}} \circ A_{\frac{\tau}{2}}]^n \begin{bmatrix} u_0(x) \\ U(u_0(x)) \end{bmatrix}. \tag{2.9}$$

In the following two subsections we are going to describe, respectively, the numerical schemes approximating the operators A_τ and B_τ.

2.3. Numerical scheme for A_τ

The numerical scheme for the conservation part (2.5) is of the Godunov type ([9,10]); however, different from ordinary Godunov-type schemes, our scheme involves two conservative entities, the numerical momentum u_j^n and numerical energy U_j^n. The scheme proceeds in reconstruction, evolution and cell-averaging steps in each time step as the following.

Reconstruction. We use the numerical solution $\{(u_j^n, U_j^n)\}$ to reconstruct a piecewise linear function $R(x;u^n,U^n)$, which is of the form

$$R(x;u^n,U^n) = u_j^n + s_j^n(x - x_j), \quad x_{j-1/2} < x < x_{j+1/2} \tag{2.10}$$

in each grid cell $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$. Obviously, $R(x;u^n,U^n)$ satisfies

$$\frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} R(x;u^n,U^n)\mathrm{d}x = u_j^n. \tag{2.11}$$

We require

$$\frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} U(R(x;u^n,U^n))\mathrm{d}x = U_j^n; \tag{2.12}$$

i.e. the energy cell-average of the reconstructed solution be equal to the numerical energy in the cell. Eq. (2.12) is an equation of slope s_j^n, from which we solve out s_j^n for the reconstruction (2.10).

Evolution. Evolve the equation with the reconstructed solution $R(x;u^n,U^n)$ as the initial function at time t_n

$$\begin{cases} v_t + f(v)_x = 0, & -\infty < x < \infty, \quad t_n < t \leqslant t_{n+1}, \\ v(x,t_n) = R(x;u^n,U^n), & -\infty < x < \infty, \end{cases} \tag{2.13}$$

and obtain the solution $v(x,t)$ over the time interval $[t_n,t_{n+1}]$.

Cell-averaging. Cell-average $v(x,t_{n+1})$ over the jth cell to obtain the numerical momentum u_j^{n+1} at t_{n+1}; i.e.

$$u_j^{n+1} = \frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} v(x,t_{n+1})\mathrm{d}x. \tag{2.14}$$

In practice, we use the integral form of the first equation in Eq. (2.5) over the cell instead to compute u_j^{n+1}

$$u_j^{n+1} = u_j^n - \lambda\left(\hat{f}^n_{j+\frac{1}{2}} - \hat{f}^n_{j-\frac{1}{2}}\right), \tag{2.15}$$

where $\lambda = \frac{\tau}{h}$ is the mesh ratio and the numerical momentum flux $\hat{f}^n_{j\pm\frac{1}{2}}$ is defined as

$$\hat{f}^n_{j\pm\frac{1}{2}} = \frac{1}{\tau}\int_{t_n}^{t_{n+1}} \frac{1}{2}\left(v\left(x_{j\pm\frac{1}{2}}, t\right)\right)^2 \mathrm{d}t. \tag{2.16}$$

The numerical energy U_j^{n+1} at t_{n+1} is computed as

$$U_j^{n+1} = U_j^n - \lambda\left(\widehat{F}^n_{j+\frac{1}{2}} - \widehat{F}^n_{j-\frac{1}{2}}\right), \tag{2.17}$$

where the numerical energy flux $\widehat{F}^n_{j\pm\frac{1}{2}}$ is defined as

$$\widehat{F}^n_{j\pm\frac{1}{2}} = \frac{1}{\tau}\int_{t_n}^{t_{n+1}} \frac{2}{3}\left(v\left(x_{j\pm\frac{1}{2}}, t\right)\right)^3 \mathrm{d}t. \tag{2.18}$$

Thus, we complete a step of computation. For stability of the scheme the mesh ratio is restricted by the CFL condition

$$\lambda \max_j\{|u_j^n|\} < 1, \tag{2.19}$$

so that waves emanating from $x_{j+\frac{1}{2}}$ at t_n will not affect the neighboring cell-edges.

Remark 2.1. One may think of computing the numerical energy at t_{n+1} as the energy cell-average of the $v(x,t)$, i.e.

$$\widetilde{U}_j^{n+1} = \frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} U(v(x, t_{n+1}))\mathrm{d}x. \tag{2.20}$$

However, computed in such a way $\widetilde{U}_j^{n+1}$ is different from U_j^{n+1} and we have from the entropy condition ([9,10])

$$\widetilde{U}_j^{n+1} \leqslant U_j^{n+1}. \tag{2.21}$$

Thus, computing U_j^{n+1} as Eq. (2.20) will not conserve the energy.

Remark 2.2. In practice, the momentum and energy fluxes (2.16) and (2.18) are evaluated approximately following the procedure described in [7]. That is, we use the mid-point rule as the numerical quadrature to evaluate the fluxes (2.16) and (2.18)

$$\hat{f}^n_{j+\frac{1}{2}} \simeq \frac{1}{2}\left(v\left(x_{j+\frac{1}{2}}, t_{n+\frac{1}{2}}\right)\right)^2 \tag{2.22}$$

and

$$\widehat{F}^n_{j+\frac{1}{2}} \simeq \frac{2}{3}\left(v\left(x_{j+\frac{1}{2}}, t_{n+\frac{1}{2}}\right)\right)^3 \tag{2.23}$$

where the predicted values for $v(x_{j+\frac{1}{2}}, t_{n+\frac{1}{2}})$ are evaluated following the so-called local Cauchy–Kowalevski procedure. Firstly, we need to compute $v(x_{j+\frac{1}{2}\pm 0}, t_{n+\frac{1}{2}})$. By the Taylor expansion of $v(x_{j+\frac{1}{2}\pm 0}, t_{n+\frac{1}{2}})$ at $(x_{j\pm\frac{1}{2}}, t_n)$ and noting the Eq. (2.5) and the reconstruction (2.10) we have

$$\begin{aligned} v\left(x_{j+\frac{1}{2}\pm 0}, t_{n+\frac{1}{2}}\right) &= v\left(x_{j+\frac{1}{2}\pm 0}, t_n\right) + \frac{\tau}{2} v_t\left(x_{j+\frac{1}{2}\pm 0}, t_n\right) + \mathrm{O}(\tau^2) \\ &= v\left(x_{j+\frac{1}{2}\pm 0}, t_n\right) - \frac{\tau}{2}\left\{\frac{1}{2}\left(v\left(x_{j+\frac{1}{2}\pm 0}, t_n\right)\right)^2\right\}_x + \mathrm{O}(\tau^2) \\ &= R\left(x_{j+\frac{1}{2}\pm 0}; u^n, U^n\right) - \frac{\tau}{2} R\left(x_{j+\frac{1}{2}\pm 0}; u^n, U^n\right) s^n_{j+\frac{1}{2}\pm\frac{1}{2}} + \mathrm{O}(\tau^2). \end{aligned} \tag{2.24}$$

Then $v(x_{j+\frac{1}{2}\pm 0}, t_{n+\frac{1}{2}})$ are computed by ignoring the error $\mathrm{O}(\tau^2)$ in Eq. (2.24). Once $v(x_{j+\frac{1}{2}\pm 0}, t_{n+\frac{1}{2}})$ are computed, the predicted value $v(x_{j+\frac{1}{2}}, t_{n+\frac{1}{2}})$ is obtained by solving the Riemann problem $\mathrm{Rie}(v(x_{j+\frac{1}{2}-0}, t_{n+\frac{1}{2}}), v(x_{j+\frac{1}{2}+0}, t_{n+\frac{1}{2}}))$.

Remark 2.3. Obviously, the designed scheme is an entropy-conservative scheme; however, different from the entropy-conservative schemes presented in [19], the entropy conservation of our scheme is not as customarily in the sense that $U(u_j^n)$ is conserved, but rather in the sense that U_j^n is conserved with U_j^n related to u_j^n as in Eqs. (2.10) and (2.12).

2.4. Numerical scheme for B_τ

The momentum equation of the dispersion part, i.e. the first equation in Eq. (2.6), is linear; therefore, it is easy to see that sliding cell-average of a solution $u(x,t)$

$$\bar{u}(x,t) = \frac{1}{h}\int_{x-\frac{h}{2}}^{x+\frac{h}{2}} u(\xi, t)\mathrm{d}x, \tag{2.25}$$

is also a solution to the equation. So is $U(\bar{u}(x,t))$ to the energy equation, the second one in Eq. (2.6). Note that the energy cell-average

$$\overline{U}(u(x,t)) = \frac{1}{h}\int_{x-\frac{h}{2}}^{x+\frac{h}{2}} u^2(\xi, t)\mathrm{d}x$$

is only $\mathrm{O}(h^2)$ away from $U(\bar{u}(x,t))$

$$\overline{U}(u(x,t)) = U(\bar{u}(x,t)) + \mathrm{O}(h^2), \tag{2.26}$$

therefore, by noting the differentiability of the error coefficient in Eq. (2.26) with respect to t, we have

$$\overline{U}_t + \varepsilon(2\bar{u}\bar{u}_{xx} - (\bar{u}_x)^2)_x = \mathrm{O}(h^2). \tag{2.27}$$

This indicates that the point-value scheme for Eq. (2.6) can also be used to compute its cell-average approximations maintaining the second-order accuracy.

We use the direct difference method in [8, Section 2.2.1] for computing the numerical momentum in B_τ. The scheme for momentum can then be stated as

$$u_j^{n+1} = u_j^n - \varepsilon\frac{\lambda}{h^2}\left(\Delta_+\Delta_- u_{j+\frac{1}{2}}^{n+\frac{1}{2}} - \Delta_+\Delta_- u_{j-\frac{1}{2}}^{n+\frac{1}{2}}\right), \tag{2.28}$$

where

$$\begin{aligned} u_{j+\frac{1}{2}}^{n+1} &= \frac{1}{2}(u_{j+1}^{n+1} + u_j^{n+1}), \quad u_{j+\frac{1}{2}}^{n} = \frac{1}{2}(u_{j+1}^{n} + u_j^{n}), \\ u_j^{n+\frac{1}{2}} &= \frac{1}{2}(u_j^n + u_j^{n+1}), \quad u_{j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}\left(u_{j+\frac{1}{2}}^{n} + u_{j+\frac{1}{2}}^{n+1}\right). \end{aligned} \tag{2.29}$$

The difference operators Δ- and Δ_+ stand for

$$\Delta_- w_\sigma = w_\sigma - w_{\sigma-1} \quad \text{and} \quad \Delta_+ w_\sigma = w_{\sigma+1} - w_\sigma, \tag{2.30}$$

respectively, with σ being an integer, say j, or a half integer, say $j+\frac{1}{2}$. The scheme (2.28) is conservative. The advantage of using this scheme, as stated in [8], is that the time step τ does not suffer from a prohibiting restriction $\tau = \mathrm{O}(h^3)$ and the corresponding linear algebra system is still easy to be solve using well developed numerical methods. Thus, the only stability limit on the time step τ in our method is the CFL condition (2.19).

The numerical energy U_j^{n+1} is computed passively in the scheme from the computed numerical momentum. To compute it, we discretize $(\bar{u}_x)^2$ and $\bar{u}\bar{u}_{xx}$ in Eq. (2.27) as

$$(\bar{u}_x)^2 \simeq \frac{1}{2}\left(\left(\Delta_+ u_{j\pm\frac{1}{2}}^{n+1/2}\right)^2 + \left(\Delta_- u_{j\pm\frac{1}{2}}^{n+1/2}\right)^2\right) \quad \text{and} \quad \bar{u}\bar{u}_{xx} \simeq u_{j\pm\frac{1}{2}}^{n+1/2}\Delta_+\Delta_- u_{j\pm\frac{1}{2}}^{n+1/2}, \tag{2.31}$$

respectively, and ignore the O(h^2) error in Eq. (2.27). The scheme for computing the numerical energy is then stated as

$$\begin{aligned} U_j^{n+1} = U_j^n &- 2\varepsilon\frac{\lambda}{h^2}\left(u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\left(\Delta_+\Delta_- u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right) - u_{j-\frac{1}{2}}^{n+\frac{1}{2}}\left(\Delta_+\Delta_- u_{j-\frac{1}{2}}^{n+\frac{1}{2}}\right)\right) + \varepsilon\frac{\lambda}{2h^2}\Delta_+\left(\left(\Delta_- u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right)^2 + \left(\Delta_- u_{j-\frac{1}{2}}^{n+\frac{1}{2}}\right)^2\right) \\ &- \varepsilon\frac{\lambda}{8h^2}\left(\left(\Delta_+\Delta_- u_{j+1}^{n+\frac{1}{2}}\right)^2 - \left(\Delta_+\Delta_- u_{j-1}^{n+\frac{1}{2}}\right)^2\right). \end{aligned} \tag{2.32}$$

The term $\frac{\lambda}{8h^2}((\Delta_+\Delta_- u_{j+1}^{n+\frac{1}{2}})^2 - (\Delta_+\Delta_- u_{j-1}^{n+\frac{1}{2}})^2)$ in Eq. (2.32) is of O(h^3) and will not affect the second-order accuracy of the scheme, which will be seen in the discussion in Section 4. The importance of this term will be seen in the proof of the following lemma presented in Appendix. The scheme (2.32) is also conservative.

Lemma 2.1. *For scheme (2.28) the following equality holds,*

$$\begin{aligned} (u_j^{n+1})^2 = (u_j^n)^2 &- 2\varepsilon\frac{\lambda}{h^2}\left(u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\left(\Delta_+\Delta_- u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right) - u_{j-\frac{1}{2}}^{n+\frac{1}{2}}\left(\Delta_+\Delta_- u_{j-\frac{1}{2}}^{n+\frac{1}{2}}\right)\right) \\ &+ \varepsilon\frac{\lambda}{2h^2}\Delta_+\left(\left(\Delta_- u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right)^2 + \left(\Delta_- u_{j-\frac{1}{2}}^{n+\frac{1}{2}}\right)^2\right) - \varepsilon\frac{\lambda}{8h^2}\left(\left(\Delta_+\Delta_- u_{j+1}^{n+\frac{1}{2}}\right)^2 - \left(\Delta_+\Delta_- u_{j-1}^{n+\frac{1}{2}}\right)^2\right). \end{aligned} \tag{2.33}$$

Lemma 2.1 indicates that scheme (2.28) is an energy-conserved scheme in the ordinary sense. The proof of the lemma is presented in Appendix. Subtracting Eq. (2.33) from Eq. (2.32), we arrive

$$U_j^{n+1} = (u_j^{n+1})^2 + U_j^n - (u_j^n)^2. \tag{2.34}$$

Thus, Eq. (2.34), instead of Eq. (2.32), is indeed the scheme practically used to compute the numerical energy.

3. Feasibility of the method and Jensen condition

As we have seen in the previous section, our method involves two entities, the numerical momentum u^n and numerical energy U^n, and both of them are computed conservatively, see Eqs. (2.15), (2.17), (2.28) and (2.32). We note that the numerical energy is not computationally passive in the method; it gives feedback to the solution in solution reconstruction in A_τ. With a given numerical energy the reconstruction slope s_j^n is solved out from Eq. (2.12). However, the first question that should be asked is whether Eq. (2.12) is solvable for s_j^n with a given U_j^n; otherwise, the method is not feasible.

By noting $U(u) = u^2$ and Eq. (2.10) it is not difficult to derive from Eq. (2.12)

$$(s_j^n)^2 = \frac{12(U_j^n - (u_j^n)^2)}{h^2}. \tag{3.1}$$

Therefore, Eq. (2.12) is solvable and admits two real roots for s_j^n provided

$$U_j^n \geqslant U(u_j^n) = (u_j^n)^2. \tag{3.2}$$

Since s_j^n is an approximation to $u_x(x_j,t_n)$, it should have the same sign as that of $(u_{j+1}^n - u_{j-1}^n)$; therefore, it is computed as

$$s_j^n = \mathrm{sgn}(u_{j+1}^n - u_{j-1}^n)\sqrt{\frac{12(U_j^n - (u_j^n)^2)}{h^2}}. \tag{3.3}$$

Now it is clear that the solvability of Eq. (2.12) for s_j^n and thus the feasibility of the method rely entirely on whether the inequality (3.2) holds. We call Eq. (3.2) the Jensen condition since when u_j^n and U_j^n are exactly the momentum and energy cell-averages of a solution (3.2) is just the Jensen's inequality on the solution in the grid cell.

Theorem 3.1. *Our method maintains the Jensen condition provided the condition holds at the initial time; therefore the method is feasible.*

Since the method proceeds in a splitting fashion as described in Eq. (2.8) or Eq. (2.9), to prove the theorem it is suffice to prove that both the schemes for A_τ and B_τ maintain the condition, i.e. once the condition holds at t_n it then holds also at t_{n+1}. Let us first investigate A_τ, for which we have the following lemma.

Lemma 3.1. *The numerical scheme presented in Section 2.3 for A_τ maintains the Jensen condition.*

Proof. Since the Jensen condition holds at t_n, Eq. (2.12) is solvable and thus the solution can be reconstructed as $R(x;u^n,U^n)$ at t_n. The IVP (2.13) then evolves with the time to obtain the solution $v(x,t)$ over the time interval $[t_n,t_{n+1}]$, by which u_j^{n+1} and U_j^{n+1} are evaluated as in Eqs. (2.14)–(2.18). Note that we have

$$U(u_j^{n+1}) = U\left(\frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} v(x,t_{n+1})\mathrm{d}x\right), \tag{3.4}$$

from which and the Jensen's inequality we have

$$U(u_j^{n+1}) \leqslant \widetilde{U}_j^{n+1}, \tag{3.5}$$

where $\widetilde{U}_j^{n+1}$ is defined as in Eq. (2.20). The lemma then follows from Eq. (2.21) in Remark 2.1. □

We then investigate the scheme for B_τ, for which we have the following lemma.

Lemma 3.2. *The numerical scheme presented in Section 2.4 for B_τ maintains the Jensen condition.*

Proof. The lemma is an immediate result of Eq. (2.34). □

Proof of Theorem 3.1. The theorem follows by combining the results of Lemmas 3.1 and 3.2 and noting Eqs. (2.8) and (2.9). □

Theorem 3.2. *For our numerical method,* (1) *the numerical momentum is uniformly L^2 bounded,* (2) *the numerical energy is nonnegative and* (3) *the numerical energy is uniformly L_1 bounded, provided the initial energy is L bounded and the Jensen condition holds at the time.*

Proof. According to Theorem 3.1, the Jensen condition holds all the times for the numerical solution; therefore, statement (2) follows from the condition (3.2) by noting $(u_j^n)^2 \geqslant 0$. Statement (3) follows from statement (2) and the conservation of energy; actually, the L_1 norm of the energy is a constant. Finally, statement (1) follows from the statement (3) and the Jensen condition. □

Remark 3.1. We know that when a numerical solution blows up, either its momentum or energy must be out of control, see [2,16,18,26]. Theorem 4.2 and its proof show that in our method the numerical momentum is L^2 controlled by the energy, and the energy itself is nonnegative and L_1 bounded, which indicates that the method is very stable in long-time numerical integration. The numerical examples in Section 5, especially the simulation of Zabusky–Kruskal's example, illustrate this.

Remark 3.2. Lemma 3.1 and thus Theorem 3.1 hold when the fluxes in A_τ are computed as in Eqs. (2.16) and (2.18). We do not know yet whether the lemma and thus the theorem are still true if the fluxes are computed practically as in Remark 2.2. However, our numerical experiments with the fluxes computed as in Remark 2.2 have not yet counted any violation of the Jensen condition. We thus conjecture that the lemma and theorem are still true even in this case.

4. Residual error analysis and scheme's accuracy

The truncation error of a numerical method for PDE is customarily defined as the error caused by replacing the numerical solution in the method with true solution to the PDE, see [9,10,15]. Equivalently, the residual error is the error of one step computation with the numerical solution at t_n being exact. The difference between the two errors is a factor of the time step τ. Because all the previously described numerical methods involve two numerical entities, the numerical momentum and numerical energy, the residual errors of our scheme should thus involve errors in computing the two of them. Moreover, the reconstruction slope s_j^n in scheme A_τ is an approximation to $u_x(x,t)$ and we also include the error in computing the slope.

Definition 4.1. The residual errors for the previously described numerical methods are defined as $(R_j^{n,u}, R_j^{n,U}, R_j^{n,s})^{\mathrm{T}}$ with

$$\begin{bmatrix} R_j^{n,u} \\ R_j^{n,U} \end{bmatrix} = \begin{bmatrix} \bar{u}_j^{n+1} \\ \overline{U}_j^{n+1} \end{bmatrix} - H_\tau \begin{bmatrix} \bar{u}_j^{n} \\ \overline{U}_j^{n} \end{bmatrix} \tag{4.1}$$

and

$$R_j^{n,s} = u_x(x_j, t_{n+1}) - s_j^{n+1}\left(H_\tau \begin{bmatrix} \bar{u}_j^{n} \\ \overline{U}_j^{n} \end{bmatrix} \right), \tag{4.2}$$

where $\bar{u}_j^n$ and $\overline{U}_j^n$ are, respectively, the momentum and energy cell-averages of true solution at (x_j,t_n), H_τ is the difference operators, A_τ, B_τ, $[A_\tau \circ B_\tau]$ for the Godunov's splitting and $[A_{\tau/2} \circ B_{\tau/2} \circ B_{\tau/2} \circ A_{\tau/2}]$ for the Strang's splitting, with A_τ and B_τ replaced with the numerical schemes described in Section 2. The slope s_j^{n+1} in Eq. (4.2) is solved out from Eq. (2.12), i.e. Eq. (3.3), with the computed momentum and energy cell-averages at t_{n+1}.

The inclusion of $R_j^{n,s}$ in the residual error is important. As will be seen in the following discussion, maintaining $R_j^{n,s}$ with certain order guarantees the numerical error in computing $(u^{n+2},U^{n+2})^{\mathrm{T}}$ from $(u^{n+1},U^{n+1})^{\mathrm{T}}$ being of the same order as that in computing $(u^{n+1},U^{n+1})^{\mathrm{T}}$ from $(u^n,U^n)^{\mathrm{T}}$. Thus, if the numerical error is accumulated in a linear fashion, the convergence of numerical solution then follows.

In the following we are going to study the accuracies of the schemes for A_τ and B_τ and the Godunov's and the Strang's splittings of our method. We first have the following theorem for the scheme for A_τ.

Theorem 4.1. *The difference scheme described in Section 2.3 for A_τ is second-order accurate away from extremes of solution in the sense that $(R_j^{n,u}, R_j^{n,U})^T = \mathrm{O}(h^3)$ and $R_j^{n,s} = \mathrm{O}(h)$. Near extremes of solution the scheme is still at least first-order accurate in that $(R_j^{n,u}, R_j^{n,U})^T = \mathrm{O}(h^2)$.*

To prove the theorem we need first to prove the following lemma.

Lemma 4.1. *If the numerical momentum u_j^n and energy U_j^n are third-order accurate*

$$u_j^n = \bar{u}_j^n + \mathrm{O}(h^3), \quad U_j^n = \overline{U}_j^n + \mathrm{O}(h^3), \tag{4.3}$$

the reconstruction slope solved from Eq. (2.12) or Eq. (3.3) is first-order accurate

$$u_x(x_j, t_n) = s_j^n\left(\begin{bmatrix} u_j^n \\ U_j^n \end{bmatrix} \right) + \mathrm{O}(h), \tag{4.4}$$

away from extremes of solution.

This result was first given and proved in [13] and later also in [12,21]. For the completion of discussion we provide the proof here.

Proof. We shall first prove the reconstruction slope solved with the true cell-averages are second-order accurate,

$$u_x(x_j, t_n) = s_j^n\left(\begin{bmatrix} \bar{u}_j^n \\ \overline{U}_j^n \end{bmatrix} \right) + \mathrm{O}(h^2), \tag{4.5}$$

away from extremes. To this end, we note the Taylor expansion of $u(x,t)$ at (x_j,t_n),

$$u(x, t_n) = u(x_j, t_n) + u_x(x_j, t_n)(x - x_j) + \frac{1}{2}u_{xx}(x_j, t_n)(x - x_j)^2 + \frac{1}{6}u_{xxx}(x_j, t_n)(x - x_j)^3 + \mathrm{O}(h^4). \tag{4.6}$$

Cell-averaging the two sides of Eq. (4.6) we obtain

$$u(x_j,t_n)=\bar{u}_j^n-\frac{1}{24}h^2u_{xx}(x_j,t_n)+\mathrm{O}(h^4), \tag{4.7}$$

Now we substitute Eq. (4.7) back into Eq. (4.6) and obtain

$$u(x,t_n)=\bar{u}_j^n-\frac{1}{24}h^2u_{xx}(x_j,t_n)+k_j(x,t_n), \tag{4.8}$$

where

$$k_j(x,t_n)=u_x(x_j,t_n)(x-x_j)+\frac{1}{2}u_{xx}(x_j,t_n)(x-x_j)^2+\frac{1}{6}u_{xxx}(x_j,t_n)(x-x_j)^3+\mathrm{O}(h^4). \tag{4.9}$$

We note

$$\overline{U}_j^n=\frac{1}{h}\int_{x_{j-1/2}}^{x_{j+1/2}}U(u(x,t_n))\mathrm{d}x=\frac{1}{h}\int_{x_{j-1/2}}^{x_{j+1/2}}(u(x,t_n))^2\mathrm{d}x. \tag{4.10}$$

Substituting Eq. (4.8) into Eq. (4.10), we have

$$\overline{U}_j^n=\frac{1}{h}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}}u^2(x)\mathrm{d}x=U(\bar{u}_j)+\frac{h^2}{12}(u'(x_j))^2+\mathrm{O}(h^4), \tag{4.11}$$

Substituting Eq. (4.11) into Eq. (3.3), we arrive

$$s_j^n\left(\begin{bmatrix}\bar{u}_j^n\\ \overline{U}_j^n\end{bmatrix}\right)=\mathrm{sgn}(\bar{u}_{j+1}^n-\bar{u}_{j-1}^n)\sqrt{(u_x(x_j,t_n))^2+\mathrm{O}(h^2)}. \tag{4.12}$$

Thus, Eq. (4.5) follows easily from Eq. (4.12).

Now if the numerical momentum and energy are third-order accurate as in Eq. (4.3), by following the same argument, we arrive

$$s_j^n\left(\begin{bmatrix}u_j^n\\ U_j^n\end{bmatrix}\right)=\mathrm{sgn}(u_{j+1}^n-u_{j-1}^n)\sqrt{(u_x(x_j,t_n))^2+\mathrm{O}(h)}. \tag{4.13}$$

Thus, the lemma follows easily from Eq. (4.13). □

Proof of Theorem 4.1. We start with the reconstruction step in A_τ. According to Eq. (4.12), the reconstruction solution (2.10) is second-order accurate. The evolution step does not cause any error; therefore, the numerical fluxes (2.16) and (2.18) are also second-order accurate, i.e.

$$\hat{f}_{j+1/2}^n=\frac{1}{\tau}\int_{t_n}^{t_{n+1}}\frac{1}{2}(u(x_{j+1/2},t))^2\mathrm{d}t+\mathrm{d}_f^n(x_{j+1/2})h^2+\mathrm{O}(h^3) \tag{4.14}$$

and

$$\widehat{F}_{j+1/2}^n=\frac{1}{\tau}\int_{t_n}^{t_{n+1}}\frac{2}{3}(u(x_{j+1/2},t))^3\mathrm{d}t+\mathrm{d}_F^n(x_{j+1/2})h^2+\mathrm{O}(h^3) \tag{4.15}$$

where $\mathrm{d}_f^n(x)$ and $\mathrm{d}_F^n(x)$ are Lipschitz continuous with respect to x. Thus, the result $(R_j^{n,u},R_j^{n,U})^{\mathrm{T}}=\mathrm{O}(h^3)$ follows easily from Eqs. (2.15) and (2.17). Then the result $R_j^{n,s}=\mathrm{O}(h)$ away from extremes follows from Lemma 4.1. Thus, the proof is complete. □

Remark 4.1. It is seen from the above proof that the numerical solution at t_{n+1} computed from the exact momentum and energy cell-averages at t_n,

$$\begin{bmatrix} u_j^{n+1} \\ U_j^{n+1} \end{bmatrix} = H_\tau \begin{bmatrix} \bar{u}_j^n \\ \overline{U}_j^n \end{bmatrix}, \tag{4.16}$$

is third-order accurate everywhere, even near extremes of solution. This is because of Eq. (4.5), the reconstruction slope s_j^n solved from $\bar{u}_j^n$ and $\overline{U}_j^n$ is at least first-order accurate, which makes the reconstruction solution is second-order accurate even near extremes. However, the computation from t_{n+1} to t_{n+2} will not be that lucky. Because of the third-order error of the numerical solution at t_{n+1}, s_j^{n+1} will lose some order of accuracy near extremes, so does the reconstruction solution there at t_{n+1}. In this time step, the first-order accurate slope s_j^{n+1} away from extremes maintains the numerical solution at t_{n+2},

$$\begin{bmatrix} u_j^{n+2} \\ U_j^{n+2} \end{bmatrix} = [H_\tau \circ H_\tau] \begin{bmatrix} \bar{u}_j^n \\ \overline{U}_j^n \end{bmatrix}, \tag{4.17}$$

to be third-order accurate away from extremes. However, near extremes the numerical solution may be less accurate; nevertheless it will still be at least second-order accurate. This is why $R_j^{n,s}$, the numerical error in computing the slope, should be included in the residual error.

Remark 4.2. Theorem 4.1 still holds even when the numerical fluxes are approximately evaluated as in Eqs. (2.22) and (2.23).

Theorem 4.2. *The difference scheme described in Section 2.4 for B_τ is second-order accurate in the sense that* $(R_j^{n,u}, R_j^{n,U})^T = O(h^3)$ and $R_j^{n,s} = O(h)$.

Proof. Scheme (2.28) is obviously second-order accurate. By noting that the approximations in Eq. (2.31) are all second-order accurate, scheme (2.32) is also second-order accurate, so is the scheme (2.34) then. We thus have $(R_j^{n,u}, R_j^{n,U})^\mathrm{T} = O(h^3)$. The result $R_j^{n,s} = O(h)$ follows from Lemma 4.1. □

Theorem 4.3. *The Godunov's splitting of our method is first-order accurate in the sense that* $(R_j^{n,u}, R_j^{n,U})^T = O(h^2)$ *and the Strang's splitting of our method is second-order accurate in the sense that* $(R_j^{n,u}, R_j^{n,U})^T = O(h^3)$ *and* $R_j^{n,s} = O(h)$ *away from extremes of solution and is at least first-order accurate near extremes of solution in that* $(R_j^{n,u}, R_j^{n,U})^T = O(h^2)$.

Proof. The theorem follows easily from the same arguments as in [17] and Lemma 4.1. □

5. Numerical experiments

In this section, we are going to present several numerical examples computed by the Strang's splitting version (2.9) of our method to show the efficiency of the method. We will give particular emphasis on the stability and structure-preserving feature of our method in long-time numerical integrations, especially in the "convection-dominated" cases. In all the examples, we use blue solid lines to represent numerical solutions and use red dot–dash lines to represent exact solutions.[1]

We start with the normal case,

$$u_t + \left(\frac{1}{2}u^2\right)_x + u_{xxx} = 0, \tag{5.1}$$

where the dispersion coefficient in Eq. (1.1) is taken to be $\varepsilon = 1$.

Example 5.1. Consider the one-soliton solution to Eq. (5.1)

$$u(x,t) = A\,\mathrm{sech}^2(\kappa x - \omega t - x_0), \tag{5.2}$$

[1] For interpretation of color in Figs. 2–12, the reader is referred to the web version of this article.

where

$$A = 12\kappa^2, \quad \omega = 4\kappa^3, \tag{5.3}$$

and x_0 is an arbitrary constant. We take $\kappa = 0.3$ and $x_0 = 0$ and take $u(x,0)$ as the initial condition.

Firstly, we use this solution to check the accuracy and convergence rate of the method. The solution region is taken to be $(-10, 10)$. To make the numerical simulation accurate we provide the boundary conditions at $x = \pm 10$ with the exact solution. The mesh ration $\lambda = \frac{\tau}{h}$ is taken to be 0.5. We perform the computation on the grids of 100, 200, 400, 800 and 1600 cells, respectively, up to $t = 10$ and the L^1, L^2 and L^∞ errors and the corresponding convergence rates of the momentum and energy are presented in Tables 1 and 2. We can clearly see from the tables that our method is second-order accurate. According to the discussion in Section 4, the accuracy of the method degenerates at the extreme of the solution and will be less than second-order there. However, the L^∞ errors and the corresponding convergence rate indicate that the accuracy degeneration at the extreme is very mild.

Secondly, we use this solution to test the stability and structure-preserving feature of the method in long-time integration. To this end, we still take the solution region to be $(-10, 10)$, however, with periodic boundary conditions at the two ends. Note that the soliton has an advection speed of *0.36*; therefore, if the periodic boundary conditions have little effect on the soliton, which decays rapidly approaching the two ends, we expect the solution assumes the initial value after a period of $500/9 \simeq 55.5$. We thus use this observation to make comparison between the numerical solution and the "nearly" exact solution, the initial value, at times of certain periods. We will use this observation also in some of the following examples to make comparisons between numerical and exact solutions.

We conduct the simulation on a grid of 100 cells ($h = 0.2$), up to 55.5 (one periods), 555.5 (10 periods) and 5555.5 (100 periods) and the numerical results are displayed in Fig. 2. We see from the Fig. 2 that the method is very stable in long-time simulation; the soliton is quite smooth with well-preserved shape even after 100 periods of movement. We note that the phase error becomes severe at longer times, which we believe is mainly caused by the computation of the dispersion part, in which $U(u)$ is passively computed.

Example 5.2. We consider the two-soliton solution to Eq. (5.1)

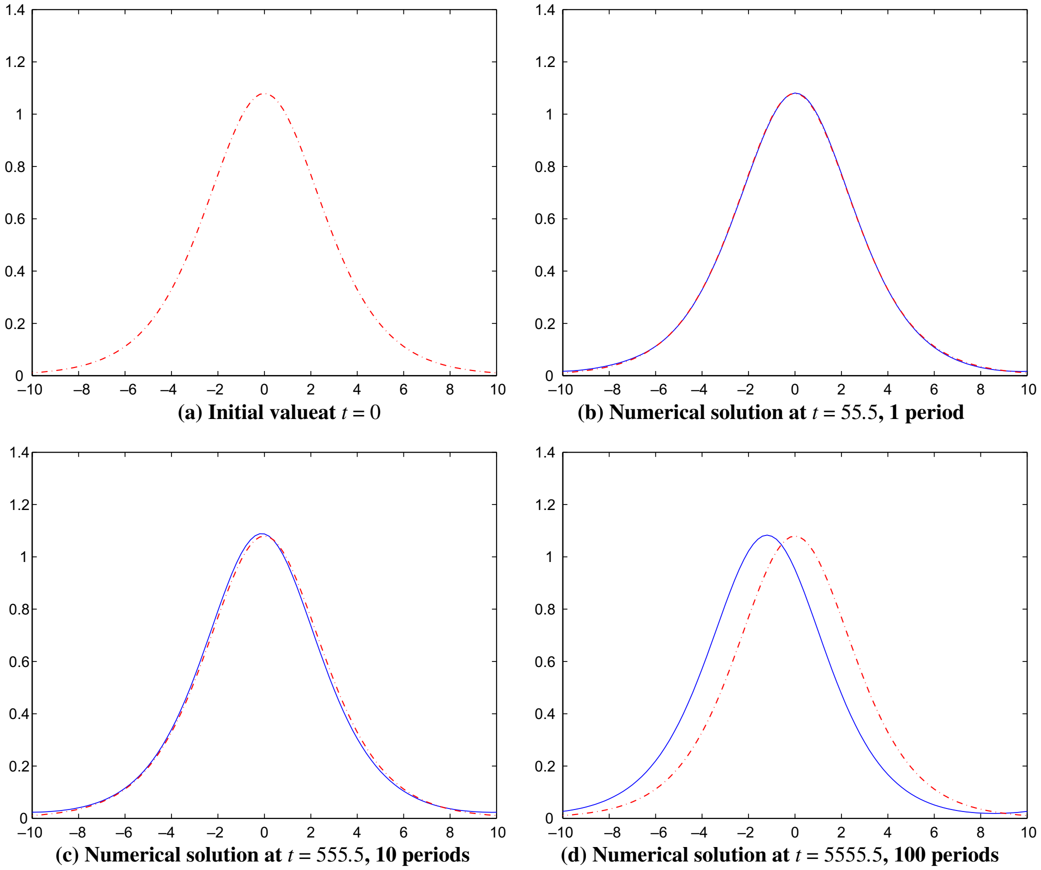
$$u(x,t) = 12\frac{\kappa_1^2 e^{\theta_1} + \kappa_2^2 e^{\theta_2} + 2(\kappa_2 - \kappa_1)^2 e^{\theta_1+\theta_2} + a^2(\kappa_2^2 e^{\theta_1} + \kappa_1^2 e^{\theta_2})e^{\theta_1+\theta_2}}{(1 + e^{\theta_1} + e^{\theta_2} + a^2 e^{\theta_1+\theta_2})^2} \tag{5.4}$$

Table 1
Example 5.1, numerical errors and convergence rates of momentum u at $t = 10$

N	h	L^∞	Order	L^2	Order	L^1	Order
100	0.2	4.606E−003	–	9.249E−003	–	3.240E−002	–
200	0.1	1.151E−003	2.001	2.304E−003	2.005	8.050E−003	2.009
400	0.05	2.880E−004	1.999	5.754E−004	2.002	2.006E−003	2.005
800	0.025	7.179E−005	2.004	1.435E−004	2.004	4.996E−004	2.005
1600	0.0125	1.768E−005	2.022	3.563E−005	2.010	1.241E−004	2.009

Table 2
Example 5.1, numerical errors and convergence rates of energy $U(u) = u^2$ at $t = 10$

N	h	L^∞	Order	L^2	Order	L^1	Order
100	0.2	7.131E−003	–	1.215E−003	–	3.074E−002	–
200	0.1	1.783E−003	2.000	3.020E−003	2.008	7.637E−003	2.009
400	0.05	4.436E−004	2.007	7.540E−004	2.002	1.909E−003	2.000
800	0.025	1.102E−004	2.009	1.882E−004	2.002	4.768E−004	2.001
1600	0.0125	2.698E−005	2.030	4.631E−005	2.023	1.178E−004	2.017



(a) Initial valueat $t = 0$

(b) Numerical solution at $t = 55.5$, 1 period

(c) Numerical solution at $t = 555.5$, 10 periods

(d) Numerical solution at $t = 5555.5$, 100 periods

Fig. 2. Numerical Example 5.1, solution plots at different times.

where

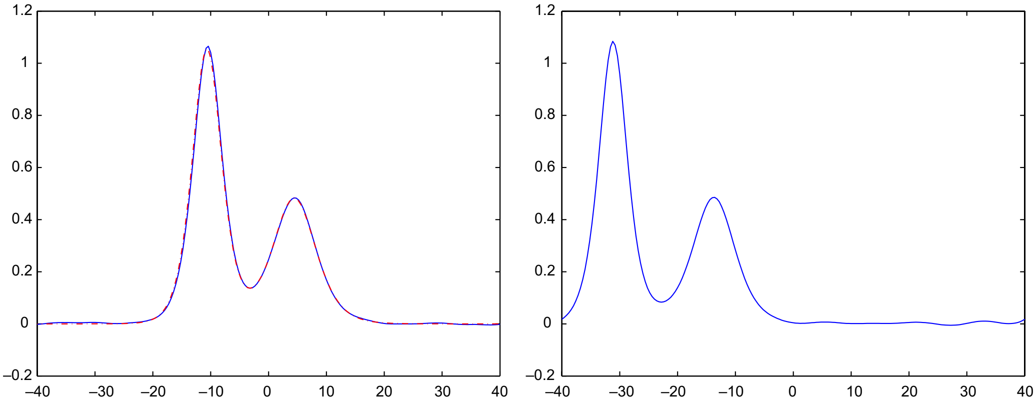
$$\kappa_1 = 0.4, \quad \kappa_2 = 0.6, \quad a^2 = \left(\frac{\kappa_1 - \kappa_2}{\kappa_1 + \kappa_2}\right)^2 = \frac{1}{25}, \tag{5.5}$$

$$\theta_1 = \kappa_1 x - \kappa_1^3 t + x_1, \quad \theta_2 = \kappa_2 x - \kappa_2^3 t + x_2, \tag{5.6}$$

$$x_1 = 4, \quad x_2 = 15. \tag{5.7}$$

To conduct the numerical simulation, we take the solution region to be $(-40, 40)$ with periodic boundary conditions at the two ends. We take $u(x,0)$ as the initial value. The grid is taken to be of 200 cells ($\Delta x = 0.4$) with the mesh ratio $\lambda = \frac{\Delta t}{\Delta x} = 0.5$. The computation is implemented up to $t = 40$ and 4000, and the plots of the numerical solution at the times are displayed in Fig. 3. We compare the numerical solution at $t = 40$ with the exact solution by ignoring the boundary boundary effects and see that the numerical solution agrees quite well with the exact one.

We note there are very weak oscillations in the numerical solution, which occur near the boundaries at the very beginning, remain in the solution and travel and interact with the solitons all the time. Because the periodic boundary conditions are implemented at the two ends, the exact solution to the problem involves a very weak discontinuity there at the beginning, which we believe is the cause of the weak oscillations. Methods with numerical viscosities (energy then diminishes), such as the one in [23,24], will immediately smoothen out the discontinuity. However, our method, preserving the conservations of momentum and energy, is sensitive to this discontinuity. Nevertheless, they are well controlled by the stability property of the method. The sensitiveness of the method to this kind of discontinuities will also be observed in the following examples, especially in the "convection-dominated" cases. However, no oscillations of this kind are observed in the



(a) Numerical solution at $t = 40$. **(b) Numerical solution at** $t = 4000$

Fig. 3. Numerical Example 5.2, solution plots at different times.

Zabusky–Kruskal's example since the periodic boundary conditions in that example do not produce discontinuity.

Now we are going to consider the "convection-dominated" cases. The following two numerical examples were used in [8] to test the performances of many numerical methods for the KdV equations. We here also use these two examples to test our numerical method, and we will emphasize on long-time integrations.

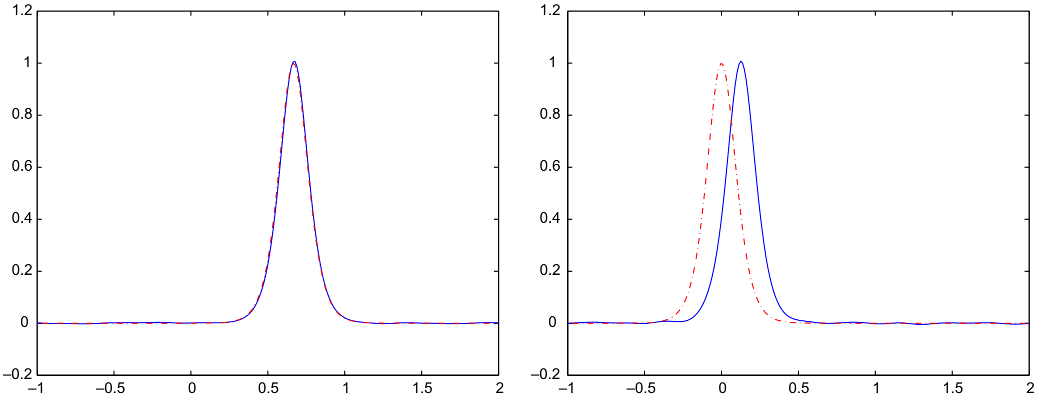
Example 5.3. We consider the KdV equation (1.1) with $\varepsilon = 0.0013020833$ and with the initial value of the one-soliton solution

$$u(x,t) = 3c\,\mathrm{sech}^2\left(\sqrt{\frac{c}{4\varepsilon}}(x-ct)\right), \tag{5.8}$$

where

$$c = 1/3. \tag{5.9}$$

The solution region is taken to be $(-1, 2)$ with periodic boundary conditions at the two ends. The solution to this problem has an advection speed of 1/3. We perform the computation on a grid of 256 (h=3/256) cells, with the mesh ratio $\lambda = \frac{\tau}{h} = 0.5$, up to $t = 2$ (1 period) and 90 (10 periods), and the plots at the times are displayed in Fig. 4. As in [8], we compare the numerical solution to the exact solution by ignoring the boundary effects.



(c) Numerical solution at $t = 2$**,1 period.** **(d) Numerical solution at** $t = 90$**,10 periods**

Fig. 4. Numerical Example 5.3, solution plots at different times, 256 points.

As is seen in the figure, the shape of the soliton is well preserved, and the numerical solution agrees quite well with the exact solution at $t = 2$. However, there is a big phase error between the two at the later time $t = 90$. The computations of this example in [8] were implemented on the grid of 64 cells. We did this with our method but the numerical result is not satisfied. Small wiggles occur in the numerical solution and thus pollute the soliton. The numerical result on the grid of 128 is OK; however, the wiggles introduced by the boundary condition become visible. It may be a shortcoming of our method that it needs more grid cells to resolve sharp soliton profiles.

Example 5.4. Consider again the KdV equation (1.1) with $\varepsilon = 0.0013020833$, however, with the initial value of the two-soliton waves

$$u(x,t) = 2\frac{\kappa_1^2 e^{\theta_1} + \kappa_2^2 e^{\theta_2} + 2(\kappa_2 - \kappa_1)^2 e^{\theta_1+\theta_2} + a^2(\kappa_2^2 e^{\theta_1} + \kappa_1^2 e^{\theta_2})e^{\theta_1+\theta_2}}{(1 + e^{\theta_1} + e^{\theta_2} + a^2 e^{\theta_1+\theta_2})^2} \tag{5.10}$$

where

$$\kappa_1 = 1, \quad \kappa_2 = 1.5, \quad a^2 = \left(\frac{\kappa_1 - \kappa_2}{\kappa_1 + \kappa_2}\right)^2 = \frac{1}{25}, \tag{5.11}$$

$$\theta_1 = \kappa_1 \frac{x}{\sqrt{6\varepsilon}} - \kappa_1^3 \frac{t}{6^{3/2}\sqrt{\varepsilon}} - 3, \tag{5.12}$$

and

$$\theta_2 = \kappa_2 \frac{x}{\sqrt{6\varepsilon}} - \kappa_2^3 \frac{t}{6^{3/2}\sqrt{\varepsilon}} + 3. \tag{5.13}$$

The solution region is taken to be $(-1, 2)$ with periodic boundary conditions at the two ends. The mesh ratio is taken to be $\lambda = \frac{\tau}{h} = 0.5$ and the computation is carried out again on grids of 256 cells up to the time $t = 4$ and 90, and the plots of the solution are displayed in Fig. 5. We compare the numerical solutions at $t = 4$ with the exact solution by ignoring the boundary effects.

Now we come to the main show of our numerical experiments, the Zabusky–Kruskal's problem.

Example 5.5. The dispersion coefficient ε in Eq. (1.1) is now 0.022^2 and the initial value is

$$u(x,0) = \cos(\pi x) \tag{5.14}$$

with periodic boundary conditions at the two ends. The solution starts with a cosine wave and later on develops a train of 8 solitons which travel at different speeds and interact with each other, see [25] for detailed

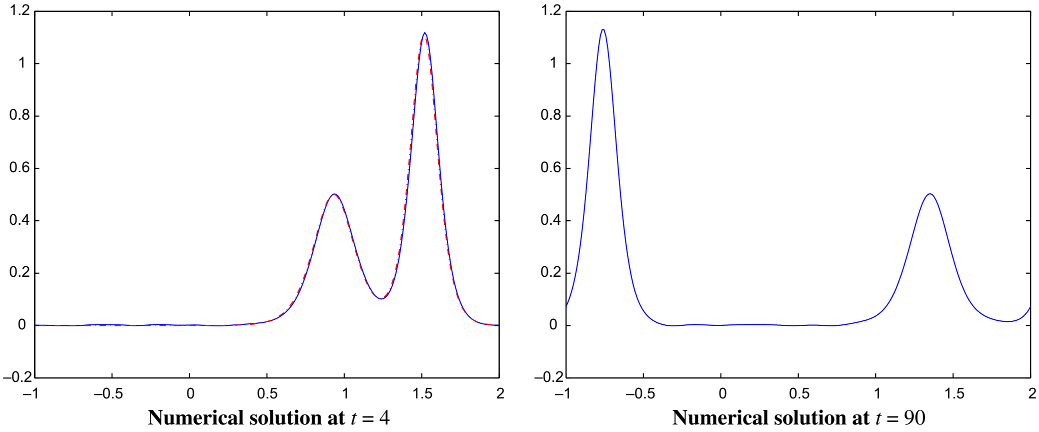
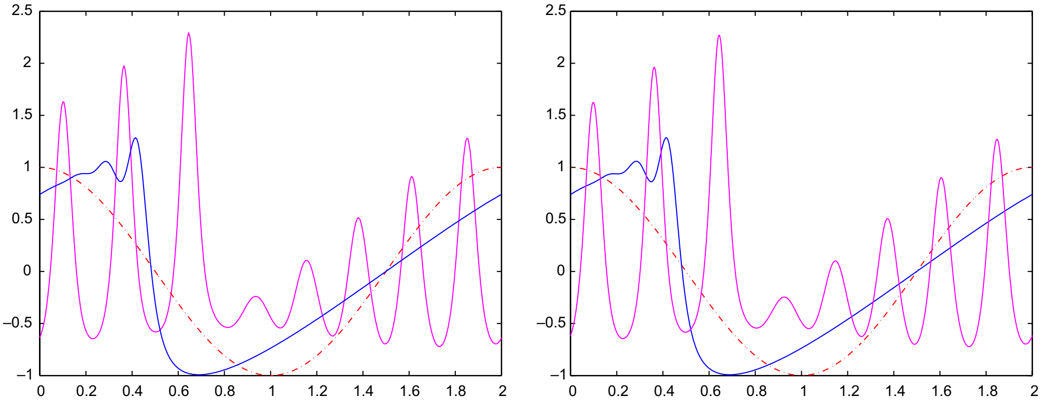


Fig. 5. Numerical Example 5.4, solution plots at different times, 256 points.

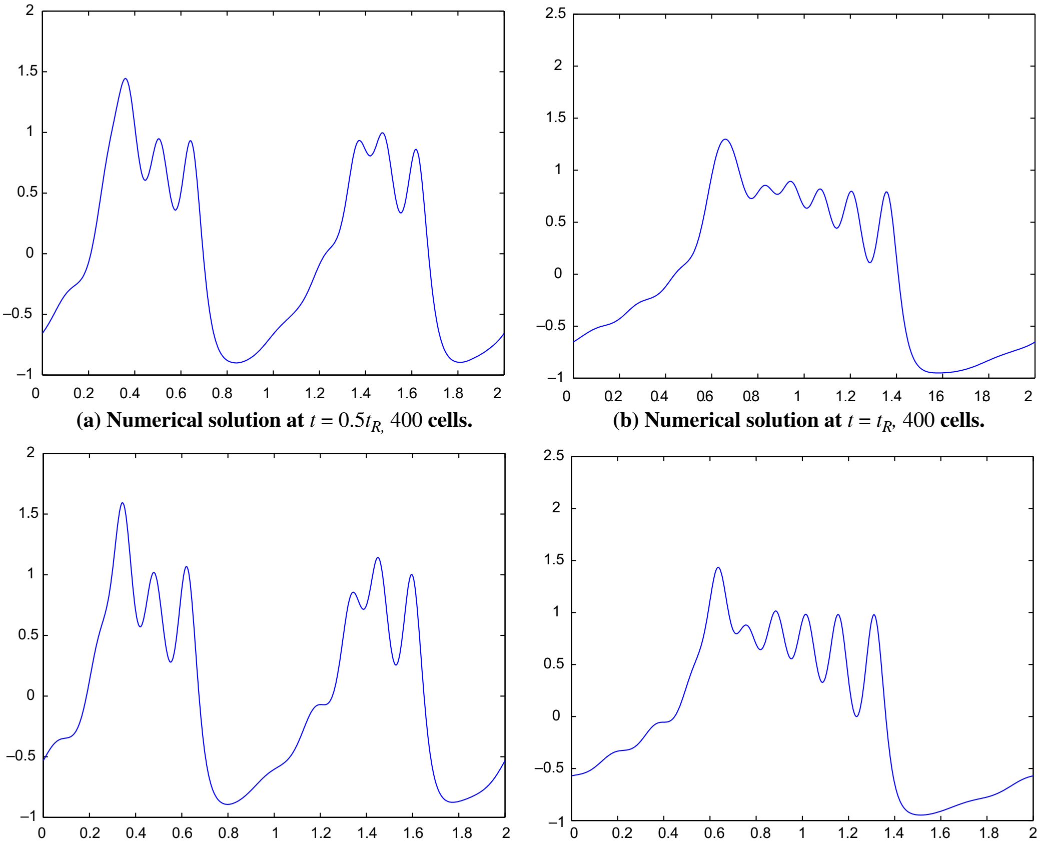
description of the solution. There are several critical moments in the development of the solution: (1) $t = t_B = \frac{1}{\pi}$ when the solution is about to breakdown, (2) $t = 3.6t_B$ when a train of 8 solitons have been developed, (3) $t = 0.5t_R = 0.5 \times 30.4t_B$ when all the odd-numbered solitons overlap at $x = 0.385$ and all the even-numbered overlap at $x = 1.385$, and (4) $t = t_R = 30.4t_B$, the recurrence time, when all the solitons arrive in almost the same phase to reconstruct the initial state.



(a) Numerical solutionon grid of 400 **cells.**

(b) Numerical solution on grid of 800 **cells.**

Fig. 6. Numerical Example 5.5, solution plots at $t = 0$, $t = t_B = \frac{1}{\pi}$ and $t = 3.6t_B$.



(a) Numerical solution at $t = 0.5t_{R,}$ 400 **cells.**

(b) Numerical solution at $t = t_R$, 400 **cells.**

(c) Numerical solution at $t = 0.5t_R$, 800 **cells.**

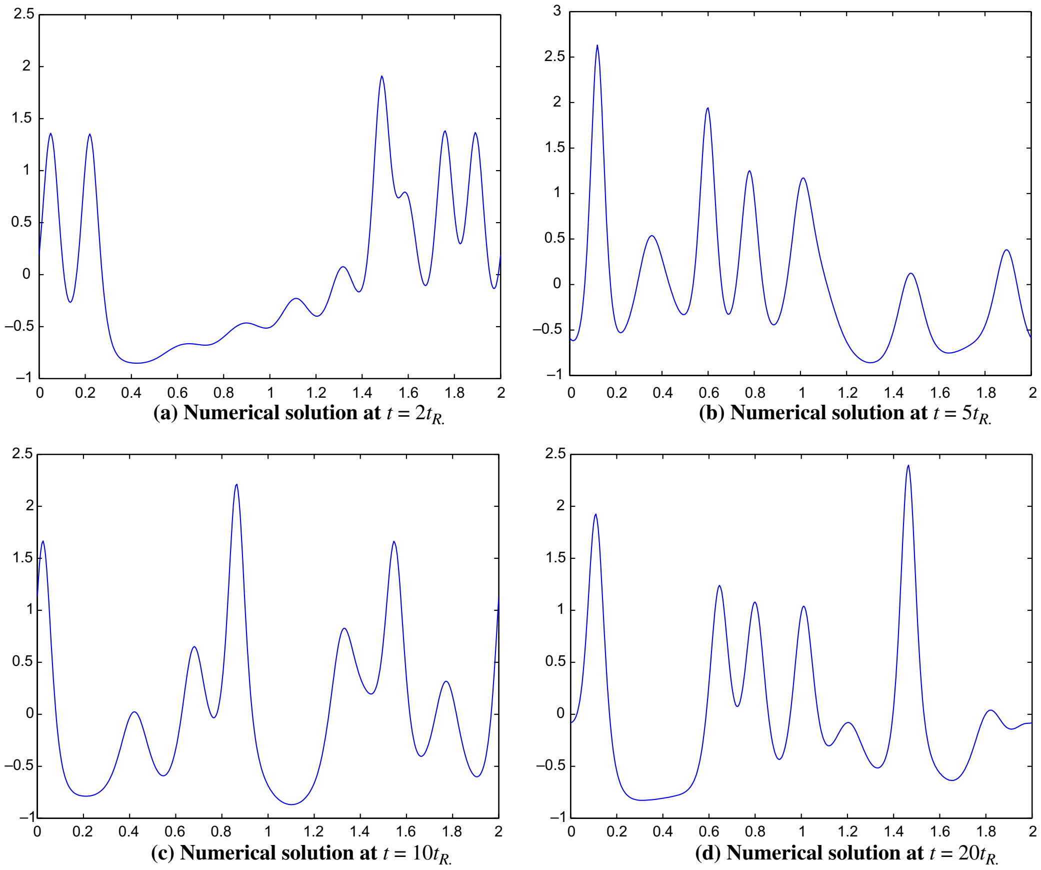
(d) Numerical solution at $t = t_R$, 800 **cells.**

Fig. 7. Numerical Example 5.5, solution plots at $t = 0.5t_R$ and $t = t_R$.

We perform our computation on grids of 400 and 800 cells, respectively, with the mesh ratio $\lambda = \frac{\tau}{h} = \frac{1}{\pi}$ as in [26]. We first carry out the computation to the time $t = t_B$ and $t = 3.6t_B$ and display the initial values and the numerical solution at the two times in Fig. 6. The two pictures in the figure have almost no difference with each other, and all of them agree quite well with the numerical results obtained by the Zabusky–Kruskal scheme, see [25], as well as the other numerical schemes, see [2,26] and [18]. This verifies the integrity of our method. We also note that there is no boundary wiggles in the numerical solution.

We then carry out the computation to the time $t = 0.5t_R$ and $t = t_R$, respectively, and the results are displayed in Fig. 7. At the recurrence time, the initial value is not well reconstructed but with oscillations. Our colleagues Deng Zhenguo and Ma Heping used their symplectic spectral method, a method with spectral discretization in space and symplectic discretization in time and with its semi-discretization satisfying three conservation relations, see [5,6], to compute the same solution at $t = t_R$, and got exactly the same results. We also note that the numerical results obtained by other methods, such as the split–step expansion schemes in [16] and the method in [18], are very similar to ours (the only exception we found is the result obtained by the Chan–Kerkhoven's method, [16]).

Most people blamed the ''over-dispersion'' features of their methods used for the simulation; however, it is hard to believe that numerical methods constructed on different philosophies and mechanism obtain numerical solutions of the same wrong. We thus conjecture that there is something missing between the KdV equation and the physical experiment conducted in laboratories, say, in Los Alamos, [25]. A numerical investigation on this matter is being carried out and we wish to report our result in due time.



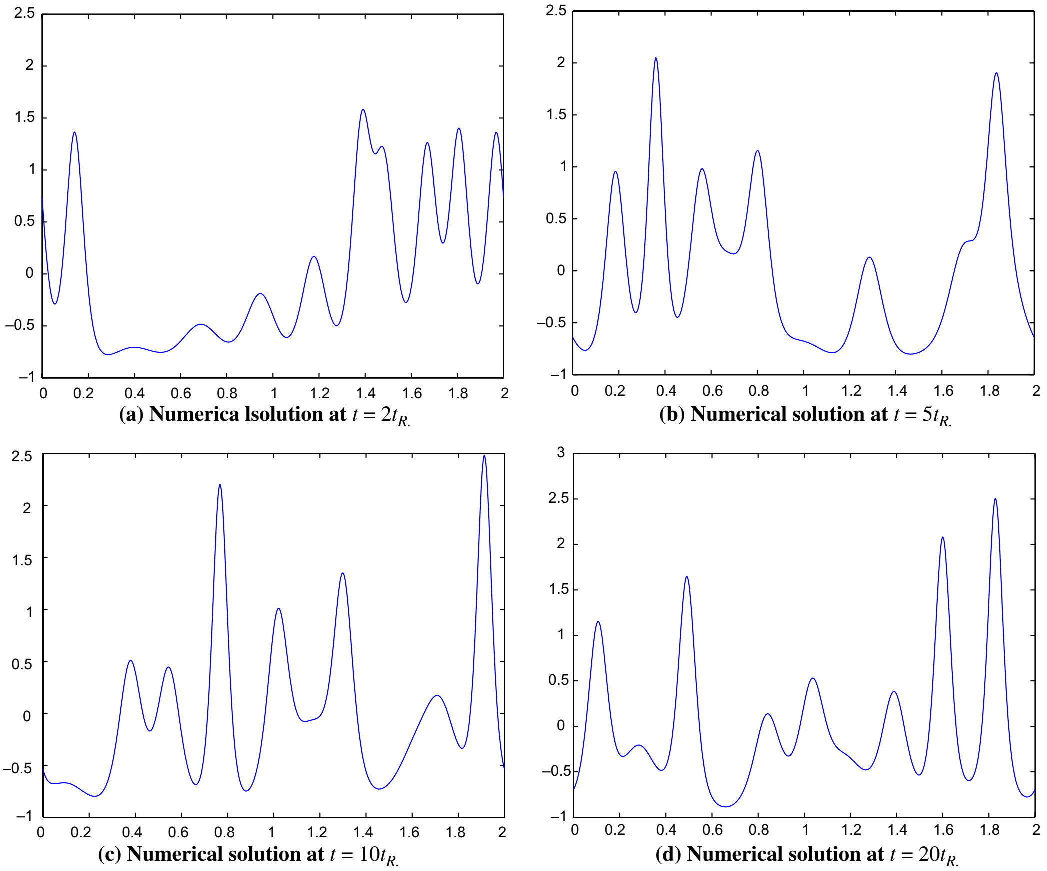
(a) Numerical solution at $t = 2t_{R.}$

(b) Numerical solution at $t = 5t_{R.}$

(c) Numerical solution at $t = 10t_{R.}$

(d) Numerical solution at $t = 20t_{R.}$

Fig. 8. Numerical Example 5.5, solution plots at $t = 2t_R$, $t = 5t_R$, $t = 10t_R$ and $t = 20t_R$, 400 cells.



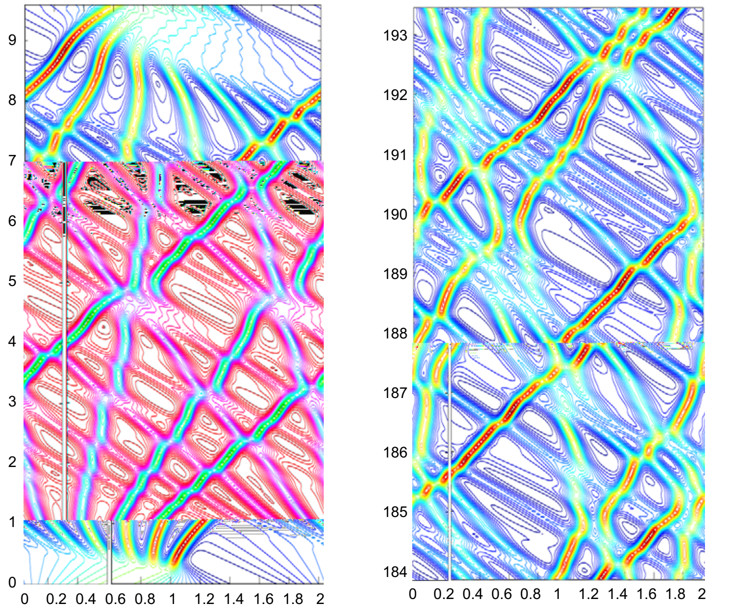
(a) **Numerica lsolution at** $t = 2t_{R.}$

(b) **Numerical solution at** $t = 5t_{R.}$

(c) **Numerical solution at** $t = 10t_{R.}$

(d) **Numerical solution at** $t = 20t_{R.}$

Fig. 9. Numerical Example 5.5, solution plots at $t = 2t_R$, $t = 5t_R$, $t = 10t_R$ and $t = 20t_R$, 800 cells.



(a) Numerical solutionin time $(0,t_R)$. **(b) Numerical solution in time** $(19t_R,20t_R)$.

Fig. 10. Numerical Example 5.5, interactions of solitons in times $(0, t_R)$ and $(19t_R, 20t_R)$, 800 cells.

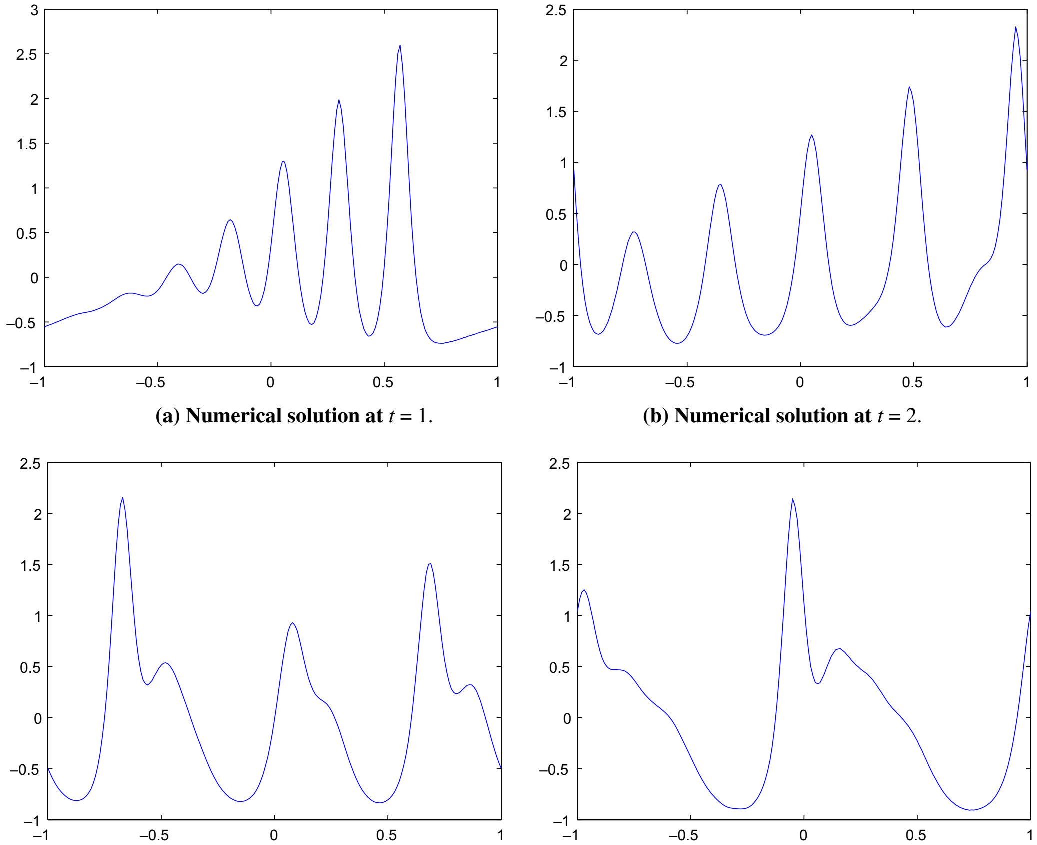
Finally, we carry out our computation to the times $t = t_R$, $t = 5t_R$, $t = 10t_R$ and $t = 20t_R$, and the numerical results on the 400-grid and 800-grid are displayed in Figs. 8 and 9, respectively. It is seen that the numerical solutions are very smooth and stable with well-preserved shapes for all the solitons. It seems to us that the computation can go on for ever without blowup. No boundary wiggles are observed in the solution. There are phase differences between the solutions on the coarse and fine grids, and we believe that the solution on the fine grid is more reliable. To show the interactions of solitons in the solution we display the (x, t)-contours of the 800-solution in the time intervals $(0, t_R)$ and $(19t_R, 20t_R)$ in Fig. 10. It is seen from the figures that there is no smearing of the solitons even at the later times.

Our scheme does not conserve the third conservation quantity $u^3 - 3\varepsilon(u_x)^2$; however, we would like to see how well the total amount of this quantity is conserved in the computation. We thus check the conservation error of this quantity at different times and the results are displayed in Table 3. In the table, the conservation error is defined as

$$\text{Error} = \sum_j \int_{x_{j-1/2}}^{x_{j+1/2}} \{R^3(x; u^n, U^n) - 3\varepsilon(s_j^n)^2\}\mathrm{d}x - \int_0^2 \{u^3(x,0) - 3\varepsilon(u_x(x,0))^2\}\mathrm{d}x, \tag{5.15}$$

Table 3
Example 5.5, conservation errors of the third conservation quantity at different times

Time	t_B	$3.6t_B$	$1t_R$	$2t_R$	$5t_R$	$10t_R$	$20t_R$
Error	4.55E−003	1.23E−001	3.99E−003	4.54E−002	4.79E−002	−6.97E−003	2.45E−002



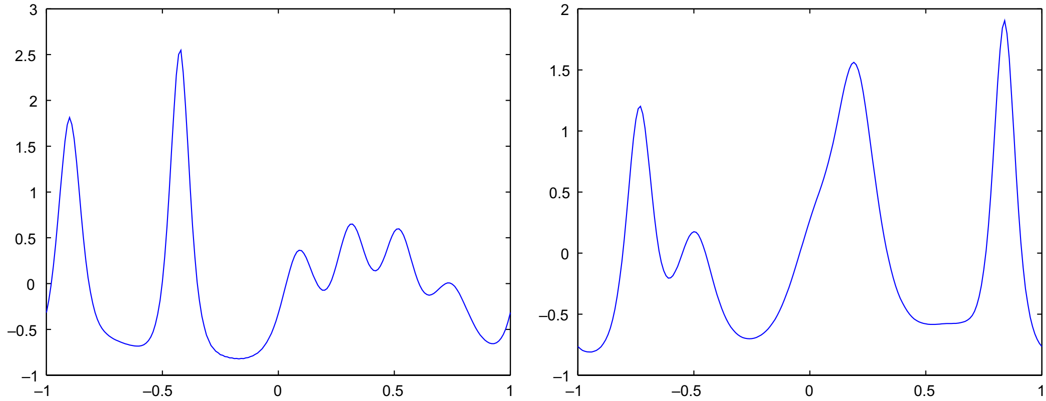
(a) Numerical solution at $t = 1$.

(b) Numerical solution at $t = 2$.

(c) Numerical solution at $t = 3$.

(d) Numerical solution at $t = 4.5$

Fig. 11. Numerical Example 5.6, solution plots at $t = 1$, $t = 2$, $t = 3$ and $t = 4.5$, 200 cells.



Numerical solution at t = 100.

Numerical solution at t = 200.

Fig. 12. Numerical Example 5.6, solution plots at $t = 100$ and $t = 200$, 200 cells.

where $R(x;u^n,U^n)$ is the reconstructed solution in the cells and s_j^n's are the slopes of the reconstructed solution. We see from the table that the third quantity is globally well conserved with an error of conservation varying between 10^{-2} and 10^{-3}. This may reflect the fact that all the solitons travel with well-preserved shapes in the numerical solution and thus the principle part of the numerical error of the solution is the phase error.

The last example is the one used in the numerical tests of [1].

Example 5.6. Consider the following generalized KdV equation

$$u_t + \left(\frac{3}{8}u^2 + \frac{1}{10}u\right)_x + \frac{2}{3} \times 10^{-3} u_{xxx} = 0, \tag{5.16}$$

with the initial value

$$u(x,0) = \cos(\pi x), \quad -1 < x < 1, \tag{5.17}$$

and periodic boundary conditions at $x = \pm 1$. We perform our computation on a grid of 200 cells ($h = 0.01$) up to the times $t = 1$, 2, 3 and 4.5 and the numerical results are displayed in Fig. 11. The results agree quite well with the ones displayed in [1]. We then carry out the computation up to the time $t = 100$ and 200 and the numerical results are displayed in Fig. 12. We see that although at these longer times, the solitons still have well-preserved shapes. It seems that the computation can go on for ever with the shapes of the solitons well preserved.

6. Conclusion

We construct a finite-volume method for the KdV equation which conserves both the momentum and energy. The main ingredient of the method is a numerical device that enables us to construct numerical schemes for a PDE that also simulate related equations. Splitting approach is employed in the construction of the method. We prove that the method is stable and second-order accurate in certain sense. Numerical experiments show that the method has a very good stability property and suits for long-time integrations of “convection-dominated” KdV equations. Phase errors in long-time integration may still be a flaw of the method. We believe that a way to ease the phase errors is to construct schemes that preserve more conservation relations.

Acknowledgements

The authors wish to thank our colleague Prof. Zhang Dajun for the enlightening discussion with him, his valuable knowledge on the KdV equation really helped us in preparing this paper. The authors wish to thank Dr. Deng Zhenguo for the numerical simulation of the Zabusky–Kruskal's example using the spectral symplectic method of Prof Ma Heping and him, the comparison of our numerical results with which verifies the integrity of our method. Finally, the authors wish to thank the referees for their appreciations of this work and their valuable comments and suggestions, which really helped us in revising the paper.

Appendix.

Proof of Lemma 2.1. Multiplying Eq. (2.28) by $2u_j^{n+\frac{1}{2}}$ we obtain

$$(u_j^{n+1})^2 = (u_j^n)^2 - 2\varepsilon\frac{\lambda}{h^2}u_j^{n+\frac{1}{2}}\left(\Delta_+\Delta_-u_{j+\frac{1}{2}}^{n+\frac{1}{2}} - \Delta_+\Delta_-u_{j-\frac{1}{2}}^{n+\frac{1}{2}}\right). \tag{A.1}$$

Subtracting Eq. (A.1) from Eq. (2.33) we obtain

$$\begin{aligned}&\left(-2u_{j+\frac{1}{2}}^{n+\frac{1}{2}} + 2u_j^{n+\frac{1}{2}} + \frac{1}{2}\Delta_+u_{j+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{1}{2}\Delta_-u_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right)\Delta_+\Delta_-u_{j+\frac{1}{2}}^{n+\frac{1}{2}} + \left(2u_{j-\frac{1}{2}}^{n+\frac{1}{2}} + 2u_j^{n+\frac{1}{2}} - \frac{1}{2}\Delta_+u_{j-\frac{1}{2}}^{n+\frac{1}{2}} + \frac{1}{2}\Delta_-u_{j-\frac{1}{2}}^{n+\frac{1}{2}}\right)\Delta_+\Delta_-u_{j-\frac{1}{2}}^{n+\frac{1}{2}}\\ &\quad - \frac{1}{8}\left(\left(\Delta_+\Delta_-u_{j+1}^{n+\frac{1}{2}}\right)^2 - \left(\Delta_+\Delta_-u_{j-1}^{n+\frac{1}{2}}\right)^2\right) = 0.\end{aligned} \tag{A.2}$$

In the derivation of Eq. (A.2) we have used the relations

$$\Delta_+\left(\Delta_-u_{j\pm\frac{1}{2}}^{n+\frac{1}{2}}\right)^2 = \left(\Delta_+u_{j\pm\frac{1}{2}}^{n+\frac{1}{2}} + \Delta_-u_{j\pm\frac{1}{2}}^{n+\frac{1}{2}}\right)\Delta_+\Delta_-u_{j\pm\frac{1}{2}}^{n+\frac{1}{2}}. \tag{A.3}$$

Thus, we need to prove Eq. (A.2), and to this end we note

$$-2u_{j+\frac{1}{2}}^{n+\frac{1}{2}} + 2u_j^{n+\frac{1}{2}} + \frac{1}{2}\Delta_+u_{j+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{1}{2}\Delta_-u_{j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{4}\Delta_+\left(\Delta_+\Delta_-u_j^{n+\frac{1}{2}}\right), \tag{A.4}$$

and

$$2u_{j-\frac{1}{2}}^{n+\frac{1}{2}} + 2u_j^{n+\frac{1}{2}} - \frac{1}{2}\Delta_+u_{j-\frac{1}{2}}^{n+\frac{1}{2}} + \frac{1}{2}\Delta_-u_{j-\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{4}\Delta_+\left(\Delta_+\Delta_-u_{j-1}^{n+\frac{1}{2}}\right), \tag{A.5}$$

by which we deduce

$$\begin{aligned}\text{LHS of Eq.(A.2)} &= \frac{1}{4}\Delta_+\left(\Delta_+\Delta_-u_j^{n+\frac{1}{2}}\right)\Delta_+\Delta_-u_{j+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{1}{4}\Delta_+\left(\Delta_+\Delta_-u_{j-1}^{n+\frac{1}{2}}\right)\Delta_+\Delta_-u_{j-\frac{1}{2}}^{n+\frac{1}{2}}\\ &\quad - \frac{1}{8}\left(\left(\Delta_+\Delta_-u_{j+1}^{n+\frac{1}{2}}\right)^2 - \left(\Delta_+\Delta_-u_{j-1}^{n+\frac{1}{2}}\right)^2\right).\end{aligned} \tag{A.6}$$

Note

$$\Delta_+\Delta_-u_{j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}\left(\Delta_+\Delta_-u_{j+1}^{n+\frac{1}{2}} + \Delta_+\Delta_-u_j^{n+\frac{1}{2}}\right), \tag{A.7}$$

and

$$\Delta_+\Delta_-u_{j-\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}\left(\Delta_+\Delta_-u_j^{n+\frac{1}{2}} + \Delta_+\Delta_-u_{j-1}^{n+\frac{1}{2}}\right), \tag{A.8}$$

equality (A.2) then follows immediately. Thus, the proof is complete. □

Remark A.1. As is seen in the above proof, the term $\left(\Delta_+\Delta_-u_{j+1}^{n+\frac{1}{2}}\right)^2 - \left(\Delta_+\Delta_-u_{j-1}^{n+\frac{1}{2}}\right)^2$ helps to make the LHS of (A.2) zero.

References

[1] U. Ascher, R. McLachlan, Multisymplectic box schemes and the Korteweg–de Vries equation, Appl. Numer. Math. 48 (2004) 255–269.

[2] U. Ascher, R. McLachlan, On symplectic and multisymplectic schemes for the KdV equation, J. Sci. Comput. 25 (2005) 83–104.

[3] T. Bridges, S. Reich, Numerical methods for Hamiltonia PDEs, J. Phys. A: Math. Gen. 39 (2006) 5287–5320.

[4] Y. Cui, Numerical scheme satisfying two conservation laws for KdV equation, Master's thesis, No. 11903–02720653, Shanghai University, 2005 (in Chinese).

[5] Z. Deng, H. Ma, Optimal error estimates of the Fourier spectral method for the KdV equation, Appl. Math. Comput., Submitted for publication.

[6] Z. Deng, Fourier spectral methods for a class of nonlocal, nonlinear dispersive wave equations, Doctoral thesis, No. 11903–04810040 Shanghai University, 2007 (in Chinese).

[7] A. Harten, B. Engquist, S. Osher, S.R. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes, III, J. Comput. Phys. 71 (1987) 231–303.

[8] H. Holden, K. Hvistendahl, N. Risebro, Operator splitting methods for generalized Korteweg–de Vries equations, J. Comput. Phys. 153 (1999) 203–222.

[9] R.J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.

[10] R.J. LeVeque, Numerical Methods for Conservation Laws, Birkhauser-Verlag, Basel, Boston, Berlin, 1990.

[11] H. Li, Z. Wang, D. Mao, Numerically neither dissipative nor compressive scheme for linear advection equation and its application to the Euler system, J. Sci. Comput., submitted for publication.

[12] H. Li, Entropy dissipating scheme for hyperbolic system of conservation laws in one space dimension, Doctoral thesis, No. 11903-02820022, Shanghai University, 2005 (in Chinese).

[13] H. Li, Second-order entropy dissipation scheme for scalar conservation laws in one space dimension, Master's thesis, No. 11903-99118086, Shanghai University, 2002 (in Chinese).

[14] H. Li, D. Mao, The design of the entropy dissipator of the entropy dissipating scheme for scalar conservation law, Chin. J. Comput. Phys. 21 (2004) 319–326, in Chinese.

[15] K. Morton, D. Mayers, Numerical Solution of Partial Differential Equations, Cambridge University Press, Britain, 2005.

[16] F. Nouri, D. Sloan, A comparison of Fourier pseudospectral methods for the solution of the Korteweg–de Vries equation, J. Comput. Phys. 83 (1989) 324–344.

[17] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (1968) 506–517.

[18] R. Takahashi, T. Ohkawa, Numerical experiment on interaction of solitons describing recurrence of initial data, Comput. Mech. 5 (1989) 273–281.

[19] E. Tadmor, Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems, Acta Numerica (2003) 451–512.

[20] Y. Wang, B. Wang, X. Chen, Multisymplectic Euler box scheme for the KdV equation, Chin. Phys. Lett. 24 (2007) 312–314.

[21] Z. Wang, Finite difference schemes satisfying multiconservation laws for linear advection equations, Master's thesis, No. 11903-99118086, Shanghai University, 2006 (in Chinese).

[22] Z. Wang, D. Mao, Conservative difference scheme satisfying three conservation laws for linear advection equation, J. SHU 12 (6) (2006) 588–592 (in Chinese).

[23] Y. Xu, C. Shu, Local discontinuous Galerkin methods for three classes of nonlinear wave equations, J. Comput. Math. 22 (2004) 250–274.

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

[25] N.J. Zabusky, M.D. Kruskal, Interactions of "solitons" in a collisionless plasma and the recurrence of initial states, Phys. Rev. Lett. 15 (1965) 240–243.

[26] P. Zhao, M. Qin, Multisymplectic geometry and multisymplectic Preissmann scheme for the KdV equation, J. Phys. A: Math. Gen. 33 (2006) 3613–3626.