Journal of Computational Mathematics
Vol.35, No.1, 2017, 52–73.
http://www.global-sci.org/jcm
doi:10.4208/jcm.1609-m2014-0069

HERMITE WENO SCHEMES WITH STRONG STABILITY PRESERVING MULTI-STEP TEMPORAL DISCRETIZATION METHODS FOR CONSERVATION LAWS*

Xiaofeng Cai
School of Mathematical Sciences, Xiamen University, Xiamen, Fujian, 361005, China
Email: xfcai89@126.com
Jun Zhu
College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, China
Email: zhujun@nuaa.edu.cn
Jianxian Qiu [1)]
School of Mathematical Sciences and Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computation, Xiamen University, Xiamen, Fujian 361005, China
Email: jxqiu@xmu.edu.cn

Abstract

Based on the work of Shu [SIAM J. Sci. Stat. Comput, 9 (1988), pp.1073-1084], we construct a class of high order multi-step temporal discretization procedure for finite volume Hermite weighted essential non-oscillatory (HWENO) methods to solve hyperbolic conservation laws. The key feature of the multi-step temporal discretization procedure is to use variable time step with strong stability preserving (SSP). The multi-step temporal discretization methods can make full use of computed information with HWENO spatial discretization by holding the former computational values. Extensive numerical experiments are presented to demonstrate that the finite volume HWENO schemes with multi-step discretization can achieve high order accuracy and maintain non-oscillatory properties near discontinuous region of the solution.

Mathematics subject classification: 65M06.
Key words: Multi-step temporal discretization; Hermite weighted essentially non-oscillatory scheme; Uniformly high order accuracy; Strong stability preserving; Finite volume scheme.

1. Introduction

In this paper, we construct a class of high order multi-step temporal discretization procedure for finite volume HWENO (Hermite weighted essential non-oscillatory) methods to solve hyperbolic conservation laws:

$$\begin{cases} u_t + \nabla \cdot F(u) = 0, \\ u(x,0) = u_0(x). \end{cases} \tag{1.1}$$

In recent years, WENO (weighted essentially non-oscillatory) schemes have been designed as a class of high order finite volume or finite difference schemes to solve hyperbolic conservation laws with the property of maintaining both uniform high order accuracy and an essentially

* Received September 28, 2014 / Revised version received March 17, 2016 / Accepted September 22, 2016 / Published online January 18, 2017 /
1) Corresponding author

non-oscillatory shock transition. WENO schemes were designed based on ENO (essentially non-oscillatory) schemes in [8,21,22]. In [12], the first WENO scheme was proposed for a third-order finite volume version in one space dimension. Third and fifth-order finite difference WENO schemes in multi-space dimensions were constructed in [9], with a general framework for the design of the smoothness indicators and nonlinear weights. Higher order finite difference WENO schemes were constructed in [1], and finite volume WENO on unstructured and structured meshes were constructed in [4,7,11,13,19]. WENO improves upon ENO in robustness, better smoothness of fluxes, better steady state convergence, better provable convergence properties, and more efficiency.

Finite volume Hermite WENO schemes were proposed in [3,14,15,26,27] and also successfully applied in solving Hamilton-Jocobi equation [16,17,28,29]. Hermite WENO schemes improve the dissipation properties of a WENO scheme due to reducing its stencil width. In fact, the compactness of a numerical stencil owes many advantages: the first, boundary conditions and complex geometries are easier to solve; the second, for the same formal accuracy, compact stencils are known to exhibit more resolution of the smaller scales by improving the dispersive and the dissipative properties of the numerical scheme, [10,23].

WENO/HWENO is a spatial discretization procedure, namely, it is a procedure to approximate the spatial derivative terms in (1.1). The time derivative term there must also be discretized. In [25], strong stability preserving (SSP) high order temporal discretizations keeping the maximum principle are used. The SSP temporal discretization methods were first proposed in [20,21], and were termed TVD (Total Variation Diminishing) temporal discretizations because the method of lines is about solving an ordinary differential equation (ODE) in time and its Euler forward version satisfy the total variation diminishing property when applied to scalar one dimensional nonlinear hyperbolic conservation laws. A class of second to fifth order SSP Runge-Kutta temporal discretizations was developed in [21]. Shu proposed a class of first order Runge-Kutta temporal discretization which have large CFL number, as well as a class of high order multi-step SSP methods in [20]. In [5], Gottlieb and Shu performed a systematic study of Runge-Kutta SSP methods, showing the optimal two stage second-order and three stage third-order SSP Runge-Kutta methods. Moreover, they proved the non-existence of four stage fourth-order SSP Runge-Kutta methods with non-negative coefficients. In [6], Gottlieb et al. reviewed and further developed SSP Runge-Kutta and multi-step methods. The new results in [6] include the optimal explicit SSP linear Runge-Kutta methods, their application to the strong stability of coercive approximations, a systematic study of explicit SSP multi-step methods, and the study of the strong stability preserving property of implicit Runge-Kutta and multi-step methods.

Multi-step temporal discretization methods can make full use of given information with spatial discretization, however, the conventional multi-step temporal discretization which is based on equal time step is not suitable for nonlinear conservation laws, for time step is variable with fixed CFL number. In this paper, we generalize the optimal few stages multi-step methods to variable size version to solve nonlinear conservation laws.

The paper is organized as follows. A brief description of the semi-discrete finite volume Hermite WENO schemes and a class of high order SSP variable step multi-step temporal discretization methods are presented in Section 2. Numerical examples are shown in the Section 3 to demonstrate the advantages of maintaining high order accuracy, the resolution and cost effective of the constructed schemes. Finally concluding remarks are given in Section 4.

2. High Order SSP Variable Step Temporal Discretization

In this section, we will first briefly review the HWENO schemes, then we present a class of high order SSP variable step multi-step temporal discretization methods.

2.1. Semi-discrete finite volume HWENO schemes

The following is the general framework of semi-discrete finite volume HWENO schemes in [14, 15, 26]. Taking the gradient of (1.1), we have

$$(\nabla u)_t^T + \nabla^T(\nabla \cdot F(u)) = 0, \tag{2.1}$$

thus

$$(\nabla u)_t^T + \nabla \cdot (\nabla \otimes F(u)) = 0. \tag{2.2}$$

For using a Hermite interpolation procedure, both the function and its derivative are needed to evolved in time. Then the governing equation of finite volume HWENO schemes is:

$$U_t + \nabla \cdot H(U) = 0, \tag{2.3}$$

where

$$U = \begin{pmatrix} u_t \\ \hline \nabla u \end{pmatrix}, \quad H(U) = \begin{pmatrix} F(u) \\ \nabla \otimes F(u) \end{pmatrix}.$$

We integrate the system (2.3) on a control volume Ω_j, which is an interval in one dimensional case or a rectangle in two dimensional cases, to obtain the semi-discrete finite volume scheme as :

$$\frac{d}{dt}\overline{U}_{\Omega_j} = -\frac{1}{|\Omega_j|}\int_{\partial\Omega_j} H(U)\cdot n ds, \tag{2.4}$$

where $|\Omega_j|$ is the volume of the control volume Ω_j. The line integral in (2.4) is typically discretized by a q-point Gaussian quadrature on each side of $\partial\Omega_j = \bigcup_{s=1}^{S}\Omega_{js}$.

$$\int_{\partial\Omega_j} H(U)\cdot n ds \approx \sum_{s=1}^{S}\Big|\partial\Omega_{js}\Big|\sum_{l=1}^{q}\omega_l H(U(G_{sl},t))\cdot n, \tag{2.5}$$

where G_{sl} and ω_l are Gaussian quadrature points and weights, respectively. The flux $H(U(G_{sl},t))\cdot n$ at Gaussian quadrature point is replaced by a numerical flux (approximate or exact Riemann solvers). For scalar equations the numerical flux can be taken as any of the monotone fluxes. For example, one could use the simple Lax-Friedrichs flux, which is given by

$$\begin{aligned} &H(U(G_{sl},t))\cdot n \\ \approx&\frac{1}{2}\Big(H(U^-(G_{sl},t)) + H(U^+(G_{sl},t))\Big)\cdot n - \alpha\Big(U^+(G_{sl},t) - U^-(G_{sl},t)\Big), \end{aligned} \tag{2.6}$$

where α is taken as an upper bound for the eigenvalues of the Jacobian in the n direction, and U^- and U^+ are the values of u inside the volume Ω_j and outside the volume Ω_j at the Gaussian point G_{sl}. The procedures of finite volume Hermite WENO reconstruction of $U^{\pm}(G_{sl},t)$ in one-dimensional and two-dimensional cases are given in detail in [14, 26], respectively. Finally, the semi-discretization HWENO scheme (2.4) can be written in the following ODE form:

$$\frac{d}{dt}\overline{U}_{\Omega_j} = L(\overline{U})_{\Omega_j}. \tag{2.7}$$

High order strong stability preserving multi-step temporal discretization methods maintaining essentially non-oscillatory property in (2.7) will be demonstrated in the following subsection.

2.2. SSP multi-step temporal discretization for nonlinear problems

We briefly review the SSP theory for explicit multi-step methods which approximate the solution of the ODE (2.7). For convenience, we rewrite (2.7) as

$$u_t = L(u), \tag{2.8}$$

with suitable initial conditions, where the spatial discretization $L(u)$ is chosen so that

$$u^{n+1} = u^n + \Delta t L(u^n), \tag{2.9}$$

satisfies the strong stability requirement $\|u^{n+1}\| \leq \|u^n\|$ in some norm $\|\cdot\|$, under the CFL condition

$$\Delta t \leq \Delta t_{FE}(u^n). \tag{2.10}$$

In [20], explicit TVD multi step methods:

$$u^{n+1} = \sum_{j=0}^{k} \Big(\alpha_i u^{n-j} + \Delta t \beta_j L(u^{n-j})\Big), \quad \alpha_j \geq 0, \quad \sum_{j=0}^{k} \alpha_j = 1 \tag{2.11}$$

are conveniently easy to manipulate into convex combinations of forward Euler steps. Just as in the Shu-Osher representation of Runge-Kutta schemes, since $\sum \alpha_j = 1$, it follows that u^{n+1} is given by a convex combination of forward Euler solvers, with suitably scaled Δt's. So such multi-step methods have the TVD property.

Some multi-step methods were proven optimal in [6]. For instance

$$u^{n+1} = \frac{16}{27}\Big(u^n + 3\Delta t L(u^n)\Big) + \frac{11}{27}\left(u^{n-3} + \frac{12}{11}\Delta t L(u^{n-3})\right) \tag{2.12}$$

with the CFL coefficient $c = \frac{1}{3}$. However, adding steps increases the CFL number, without requiring additional computation, only additional storage. For example,

$$u^{n+1} = \frac{25}{32}u^n + \frac{25}{16}\Delta t L(u^n) + \frac{7}{32}u^{n-4} + \frac{5}{16}\Delta t L(u^{n-4}), \tag{2.13}$$

$$u^{n+1} = \frac{108}{125}u^n + \frac{36}{25}\Delta t L(u^n) + \frac{17}{125}u^{n-5} + \frac{6}{25}\Delta t L(u^{n-5}), \tag{2.14}$$

though not proven optimal, increase the CFL to c=1/2 and c=0.567, respectively.

For hyperbolic conservation law, in order to keep the stability of numerical methods, the time step Δt is controlled by CFL condition, that is $\Delta t = CFL\Delta x/C$, where C is the maximum speed of wave which is depended on the solution at t_n. The maximum speed of wave is usual variable according to t_n, that is time step is variable. We will generalize high order SSP multi-step methods to nonuniform temporal step version in the next subsection.

2.2.1. A class of multi-step with variable time step and SSP discretization

Similar to (2.11), we define $(k+1)$-step finite difference method with variable time step for the first order ODE (2.8) as:

$$u^{n+1} = \sum_{j=0}^{k} \Big(\alpha_j(t_{n+1}, \cdots, t_{n-k})u^{n-j} + \beta_j(t_{n+1}, \cdots, t_{n-k})L(u^{n-j})\Big), \tag{2.15}$$

with

$$|\alpha_k(t_{n+1},\cdots,t_{n-k})|+|\beta_k(t_{n+1},\cdots,t_{n-k})|>0,$$

where t_n are temporal grid points and $\Delta t_{n-1}=t_n-t_{n-1}$, $n=1,2,\cdots$ are temporal steps.

We define the truncation error $T[u(t_n)]$ of the method (2.15)

$$T[u(t_n)]=u(t_{n+1})-\sum_{j=0}^{k}\Big(\alpha_j(t_{n+1},\cdots,t_{n-k})u(t_{n-j})+\beta_j(t_{n+1},\cdots,t_{n-k})L(u(t_{n-j}))\Big). \quad (2.16)$$

If we assume that u is sufficiently smooth, use Taylor expand we have

$$T[u(t_n)]=c_0u(t_n)+c_1u^{(1)}(t_n)+\cdots+c_qu^{(q)}(t_n)+\cdots,$$

where

$$\begin{aligned}
&c_0=1-(\alpha_0+\alpha_1+\cdots+\alpha_k),\\
&c_1=(t_{n+1}-t_n)-\sum_{j=1}^{k}\alpha_j(t_n)(t_{n-j}-t_n)-\sum_{j=0}^{k}\beta_j(t_n),\\
&\cdots\cdots\\
&c_q=\frac{1}{q!}\Big((t_{n+1}-t_n)^q-\sum_{j=1}^{k}\alpha_j(t_n)(t_{n-j}-t_n)^q\Big)-\frac{1}{(q-1)!}\sum_{j=0}^{k}\beta_j(t_n)(t_{n-j}-t_n)^{q-1}.
\end{aligned}$$

If $u(t)\in C^{p+1}(I)$, we choose k and α_j,β_j, such that $c_0=c_1=\cdots=c_p=0$ and $c_{p+1}\neq 0$, we obtain

$$T[u(t_n)]=c_{p+1}u^{(p+1)}(t_n)+O(h^{p+2}),$$

where $h=\max_{i=1}^{n}(t_i-t_{i-1})$, then the method (2.15) is called as a *linear p-order (k+1)-step method.*

2.2.2. A optimal three order variable step size SSP multi-step method

We generalize the third order SSP multi-step method (2.12) to a nonuniform temporal step 3-order 4-step method. In order to keep SSP (strong stability preserving) property, the parameters $\alpha_0,\alpha_1,\alpha_2,\alpha_3,\beta_0,\beta_1,\beta_2,\beta_3$ should satisfy:

$$\alpha_0+\alpha_1+\alpha_2+\alpha_3=1, \quad (2.17a)$$

$$\sum_{j=1}^{3}\alpha_j(t_{n-j}-t_n)-\sum_{j=0}^{3}\beta_j=t_{n+1}-t_n, \quad (2.17b)$$

$$\frac{1}{2}\sum_{j=1}^{3}\alpha_j(t_{n-j}-t_n)^2-\sum_{j=0}^{3}\beta_j(t_{n-j}-t_n)=\frac{1}{2}(t_{n+1}-t_n)^2, \quad (2.17c)$$

$$\frac{1}{6}\sum_{j=1}^{3}\alpha_j(t_{n-j}-t_n)^3-\frac{1}{2}\sum_{j=0}^{3}\beta_j(t_{n-j}-t_n)^2=\frac{1}{6}(t_{n+1}-t_n)^3. \quad (2.17d)$$

Letting $\alpha_1=\alpha_2=\beta_1=\beta_2=0$, we solve $\alpha_0,\alpha_3,\beta_0,\beta_3$ from the system (2.17) to get

$$u^{n+1}=\frac{t_{n-3}^3-3t_{n-3}t_{n+1}^2-3t_{n-3}^2t_n+6t_nt_{n-3}t_{n+1}-3t_nt_{n+1}^2+2t_{n+1}^3}{(t_{n-3}-t_n)^3}u^n$$

$$
\begin{aligned}
&- \frac{t_{n-3}^2 t_n - t_{n-3}^2 t_{n+1} + 2t_{n-3}t_{n+1}^2 - 2t_n t_{n-3} t_{n+1} + t_n t_{n+1}^2 - t_{n+1}^3}{(t_{n-3}-t_n)^2} L(u^n) \\
&+ \frac{3t_{n-3}t_n^2 - t_n^3 + 3t_{n-3}t_{n+1}^2 - 6t_n t_{n-3} t_{n+1} + 3t_n t_{n+1}^2 - 2t_{n+1}^3}{(t_{n-3}-t_n)^3} u^{n-3} \\
&- \frac{t_{n-3}t_{n+1}^2 + t_{n-3}t_n^2 - 2t_n t_{n-3} t_{n+1} - t_n^2 t_{n+1} - t_{n+1}^3 + 2t_n t_{n+1}^2}{(t_{n-3}-t_n)^2} L(u^{n-3}).
\end{aligned}
$$

Let $h_1 = \Delta t_{n-3}$, $h_2 = \Delta t_{n-2}$, $h_3 = \Delta t_{n-1}$, $h_4 = \Delta t_n$, and $\lambda_2 = \frac{h_2}{h_1}$, $\lambda_3 = \frac{h_3}{h_1}$, $\lambda_4 = \frac{h_4}{h_1}$. We obtain

$$
\begin{aligned}
u^{n+1} =& \frac{(1+\lambda_2+\lambda_3-2\lambda_4)(1+\lambda_2+\lambda_3+\lambda_4)^2}{(1+\lambda_2+\lambda_3)^3} u^n + \frac{(1+\lambda_2+\lambda_3+\lambda_4)^2}{(1+\lambda_2+\lambda_3)^2} h_4 L(u^n) \\
&+ \frac{\lambda_4^2(3+3\lambda_2+3\lambda_3+2\lambda_4)}{(1+\lambda_2+\lambda_3)^3} u^{n-3} + \frac{\lambda_4(1+\lambda_2+\lambda_3+\lambda_4)}{(1+\lambda_2+\lambda_3)^2} h_4 L(u^{n-3}).
\end{aligned} \tag{2.18}
$$

We rewrite (2.18) as:

$$
\begin{aligned}
u^{n+1} = \alpha_0(\lambda_2,\lambda_3,\lambda_4)u^n + \gamma_0(\lambda_2,\lambda_3,\lambda_4)h_4 L(u^n) + \alpha_3(\lambda_2,\lambda_3,\lambda_4)u^{n-3} \\
+ \gamma_3(\lambda_2,\lambda_3,\lambda_4)h_4 L(u^{n-3}).
\end{aligned} \tag{2.19}
$$

with

$$
\begin{aligned}
\alpha_0(\lambda_2,\lambda_3,\lambda_4) &= \frac{(1+\lambda_2+\lambda_3-2\lambda_4)(1+\lambda_2+\lambda_3+\lambda_4)^2}{(1+\lambda_2+\lambda_3)^3}, \\
\gamma_0(\lambda_2,\lambda_3,\lambda_4) &= \frac{(1+\lambda_2+\lambda_3+\lambda_4)^2}{(1+\lambda_2+\lambda_3)^2}, \\
\alpha_3(\lambda_2,\lambda_3,\lambda_4) &= \frac{\lambda_4^2(3+3\lambda_2+3\lambda_3+2\lambda_4)}{(1+\lambda_2+\lambda_3)^3}, \\
\gamma_3(\lambda_2,\lambda_3,\lambda_4) &= \frac{\lambda_4(1+\lambda_2+\lambda_3+\lambda_4)}{(1+\lambda_2+\lambda_3)^2}.
\end{aligned}
$$

Proposition 2.1. *If the forward Euler method combined with the spatial discretization L in* (2.9) *is strongly stable under the CFL restriction* (2.10),

$$
\|u^n + \Delta t_n L(u^n)\| \le \|u^n\|
$$

and $\lambda_4 \in (0, \frac{1+\lambda_2+\lambda_3}{2})$, *then the multi-step method* (2.19) *satisfies*

$$
\|u^{n+1}\| \le \max\{(\|u^n\|), (\|u^{n-3}\|)\},
$$

under the CFL restriction

$$
\Delta t_n \le \min\left\{\frac{\alpha_0}{\gamma_0}\Delta t_{FE}(u^n), \frac{\alpha_3}{\gamma_3}\Delta t_{FE}(u^{n-3})\right\}. \tag{2.20}
$$

Proof. For convenience, we write (2.19) as

$$
u^{n+1} = \alpha_0\left(u^n + \frac{\gamma_0}{\alpha_0}\Delta t_n L(u^n)\right) + \alpha_3\left(u^{n-3} + \frac{\gamma_3}{\alpha_3}\Delta t_n L(u^{n-3})\right). \tag{2.21}
$$

Since $\lambda_4 \in (0, \frac{1+\lambda_2+\lambda_3}{2})$, all coefficients α_0, γ_0, α_3 and γ_3 are greater than 0. Under the CFL restriction (2.20), we have the CFL restriction

$$0 < \frac{\gamma_0}{\alpha_0}\Delta t_n \le t_{FE}(u^n), \quad 0 < \frac{\gamma_3}{\alpha_3}\Delta t_n \le t_{FE}(u^{n-3}). \tag{2.22}$$

Then we obtain the following results under CFL restriction (2.22),

$$\left\| u^n + \frac{\gamma_0}{\alpha_0}\Delta t_n L(u^n) \right\| \le \left\| u^n \|, \| u^{n-3} + \frac{\gamma_3}{\alpha_3}\Delta t_n L(u^{n-3}) \right\| \le \| u^{n-3} \|.$$

Let $\alpha_1 = 0$ and $\alpha_2 = 0$. Then $\alpha_0 + \alpha_3 = 1$. From (2.21) we have

$$\begin{aligned} \|u^{n+1}\| \le & \alpha_0 \left(\|u^n + \frac{\gamma_0}{\alpha_0}\Delta t_n L(u^n)\| \right) + \alpha_3 \left(\|u^{n-3} + \frac{\gamma_3}{\alpha_3}\Delta t_n L(u^{n-3})\| \right) \\ \le & \alpha_0 \left(\|u^n\| \right) + \alpha_3 \left(\|u^{n-3}\| \right) \\ \le & \max\{ \left(\|u^n\| \right), \left(\|u^{n-3}\| \right) \}. \end{aligned}$$

Hence the proposition is proved. □

Remark 2.1. In order to choose time step Δt_n such that

$$\Delta t_n = \min\left\{ \frac{\alpha_0}{\gamma_0}\Delta t_{FE}(u^n), \frac{\alpha_3}{\gamma_3}\Delta t_{FE}(u^{n-3}) \right\}.$$

Let $\Delta t'_{FE} = \min\{\Delta t_{FE}(u^n), \Delta t_{FE}(u^{n-3})\}$. Then we solve Δt_n by $\frac{\Delta t_n}{\Delta t'_{FE}} = \min\{\frac{\alpha_0}{\gamma_0}, \frac{\alpha_3}{\gamma_3}\}$, i.e. solving λ_4 from

$$\lambda_4 \frac{\Delta t_{n-3}}{\Delta t'_{FE}} = \min\left\{ \frac{\alpha_0(\lambda_4)}{\gamma_0(\lambda_4)}, \ \frac{\alpha_3(\lambda_4)}{\gamma_3(\lambda_4)} \right\}. \tag{2.23}$$

Obviously, such Δt_n satisfy (2.20) since

$$\Delta t_n = \min\left\{ \frac{\alpha_0}{\gamma_0}\Delta t'_{FE}, \frac{\alpha_3}{\gamma_3}\Delta t'_{FE} \right\} \le \min\left\{ \frac{\alpha_0}{\gamma_0}\Delta t_{FE}(u^n), \frac{\alpha_3}{\gamma_3}\Delta t_{FE}(u^{n-3}) \right\}.$$

The λ_4 can be determined by (2.23). For implementation convenience, we borrow the CFL number from (2.12) and choose a smaller CFL to design $\Delta t_n = CFL\Delta t_{FE}(u^n) = CFL\Delta x/C$ (u^n), where $C(u^n)$ is the maximum speed of wave at time t^n in this paper, because this proposition generalizes the results of (2.12). Our numerical experiments in this paper show such time size Δt_n works well.

Remark 2.2. We can generalize scheme (2.13)-(2.14) to a nonuniform temporal step version, respectively.

$$\begin{aligned} u^{n+1} = & \frac{(1+\lambda_2+\lambda_3+\lambda_4-2\lambda_5)(1+\lambda_2+\lambda_3+\lambda_4+\lambda_5)^2}{(1+\lambda_2+\lambda_3+\lambda_4)^3} u^n \\ & + \frac{(1+\lambda_2+\lambda_3+\lambda_4+\lambda_5)^2}{(1+\lambda_2+\lambda_3+\lambda_4)^2} h_5 L(u^n) + \frac{\lambda_5^2(3+3\lambda_2+3\lambda_3+3\lambda_4+2\lambda_5)}{(1+\lambda_2+\lambda_3+\lambda_4)^3} u^{n-4} \\ & + \frac{\lambda_5(1+\lambda_2+\lambda_3+\lambda_4+\lambda_5)}{(1+\lambda_2+\lambda_3+\lambda_4)^2} h_5 L(u^{n-4}), \end{aligned} \tag{2.24}$$

where $h_1 = \Delta t_{n-3}$, $h_2 = \Delta t_{n-2}$, $h_3 = \Delta t_{n-1}$, $h_4 = \Delta t_n$, $h_5 = \Delta t_{n+1}$, and $\lambda_2 = \frac{h_2}{h_1}$, $\lambda_3 = \frac{h_3}{h_1}$, $\lambda_4 = \frac{h_4}{h_1}$, $\lambda_5 = \frac{h_5}{h_1}$. Moreover,

$$\begin{aligned} u^{n+1} = & \frac{(1+\lambda_2+\lambda_3+\lambda_4+\lambda_5-2\lambda_6)(1+\lambda_2+\lambda_3+\lambda_4+\lambda_5+\lambda_6)^2}{(1+\lambda_2+\lambda_3+\lambda_4+\lambda_5)^3} u^n \\ & + \frac{(1+\lambda_2+\lambda_3+\lambda_4+\lambda_5+\lambda_6)^2}{(1+\lambda_2+\lambda_3+\lambda_4+\lambda_5)^2} h_6 L(u^n) + \frac{\lambda_6^2(3+3\lambda_2+3\lambda_3+3\lambda_4+3\lambda_5+2\lambda_6)}{(1+\lambda_2+\lambda_3+\lambda_4+\lambda_5)^3} u^{n-5} \\ & + \frac{\lambda_6(1+\lambda_2+\lambda_3+\lambda_4+\lambda_5+\lambda_6)}{(1+\lambda_2+\lambda_3+\lambda_4+\lambda_5)^2} h_6 L(u^{n-5}), \end{aligned} \tag{2.25}$$

where $h_1 = \Delta t_{n-4}$, $h_2 = \Delta t_{n-3}$, $h_3 = \Delta t_{n-2}$, $h_4 = \Delta t_{n-1}$, $h_5 = \Delta t_n$, $h_6 = \Delta t_{n+1}$, and $\lambda_2 = \frac{h_2}{h_1}$, $\lambda_3 = \frac{h_3}{h_1}$, $\lambda_4 = \frac{h_4}{h_1}$, $\lambda_5 = \frac{h_5}{h_1}$, $\lambda_6 = \frac{h_6}{h_1}$.

Remark 2.3. For multi-step method, we require that the starting procedure to be the same accuracy as multi-step method and also be strong stability preserving. For example, we can use the following third order SSP Runge-Kutta method [21] to compute the starting steps with suitable reduced time steps.

$$u^{(1)} = u^n + \Delta t L(u^n), \tag{2.26a}$$

$$u^{(2)} = \frac{3}{4} u^n + \frac{1}{4}\Big(u^{(1)} + \Delta t L(u^{(1)})\Big), \tag{2.26b}$$

$$u^{n+1} = \frac{1}{3} u^n + \frac{2}{3}\Big(u^{(2)} + \Delta t L(u^{(2)})\Big). \tag{2.26c}$$

3. Numerical Results

In this section, we perform numerical experiments to test the performance of the fifth-order HWENO schemes with the third-order multi-step method (HWENO5-multi3), and compare the results with that of the fifth-order HWENO schemes with the third order SSP Runge-Kutta temporal discretization (HWENO5-RK3) for one-dimensional conservation laws. For two-dimensional conservation laws, we present the results of our numerical experiments for the fourth-order HWENO schemes with the third-order multi-step method (HWENO4-multi3), and compare the results with that of the fourth-order HWENO schemes with the third order SSP Runge-Kutta temporal discretization (HWENO4-RK3).

For CPU time comparison, the computations of one-dimensional tests are performed on a personal computer, Intel(R) Core (TM) i3-2310M CPU @ 2.10 GHz with 2.00 GB ram, while the computations of two-dimensional tests are performed on a cluster, Intel(R) Xeon(R) CPU E5405@ 2.00GHz. A uniform mesh with N cells is used for all the test cases, the CFL number is taken as 0.3 for HWENO5-multi3 and 0.2 for HWENO4-multi3, and 0.8 for HWENO5-RK3 and 0.6 for HWENO4-RK3.

3.1. One-dimensional test cases

Example 3.1. We solve the linear convection equation

$$u_t + u_x = 0, \tag{3.1}$$

with initial condition $u_0(x)$ and periodic boundary conditions.

Table 3.1: HWENO5-multi3 and HWENO5-RK3, for the linear equation with initial data $u_0(x)$. $\Delta x = \frac{1}{N}$ and $\Delta t = CFL\Delta x^{\frac{5}{3}}$.

N	HWENO5-multi3				HWENO5-RK3			
	L_1 error	Order	L_∞ error	Order	L_1 error	Order	L_∞ error	Order
	$u_0(x) = 0.5 + \sin(2\pi x)$							
10	1.58E-02		2.48E-02		1.58E-02		2.47E-02	
20	5.93E-04	4.73	1.14E-03	4.44	5.91E-04	4.74	1.14E-03	4.44
40	1.76E-05	5.08	3.39E-05	5.08	1.75E-05	5.07	3.38E-05	5.08
80	5.42E-07	5.02	1.09E-06	4.96	5.41E-07	5.02	1.09E-06	4.95
160	1.69E-08	5.01	3.29E-08	5.05	1.68E-08	5.01	3.28E-08	5.05
320	5.26E-10	5.00	9.77E-10	5.07	5.24E-10	5.00	9.73E-10	5.08
	$u_0(x) = 0.5 + \sin^4(2\pi x)$							
10	2.17E-01		3.23E-01		2.16E-01		3.23E-01	
20	4.60E-02	2.24	9.59E-02	1.75	4.58E-02	2.24	9.53E-02	1.76
40	3.74E-03	3.62	7.99E-03	3.59	3.73E-03	3.62	7.96E-03	3.58
80	3.94E-04	3.24	1.47E-03	2.44	3.91E-04	3.25	1.46E-03	2.44
160	1.20E-05	5.04	6.12E-05	4.59	1.19E-05	5.04	6.08E-05	4.59
320	3.09E-07	5.28	1.50E-06	5.35	3.06E-07	5.28	1.49E-06	5.35
	$u_0(x) = 0.5 + \sin^8(2\pi x)$							
10	2.18E-01		3.21E-01		2.17E-01		3.21E-01	
20	9.08E-02	1.26	2.65E-01	0.28	9.05E-02	1.26	2.64E-01	0.28
40	1.06E-02	3.09	3.49E-02	2.93	1.06E-02	3.10	3.46E-02	2.93
80	9.65E-04	3.46	2.34E-03	3.90	9.56E-04	3.46	2.32E-03	3.90
160	3.41E-05	4.82	1.05E-04	4.48	3.38E-05	4.82	1.04E-04	4.48
320	1.09E-06	4.96	3.74E-06	4.81	1.08E-06	4.96	3.71E-06	4.81

Three smooth initial conditions $u_0(x) = 0.5+\sin(2\pi x)$, $u_0(x) = 0.5+\sin^4(2\pi x)$ and $u_0(x) = 0.5+\sin^8(2\pi x)$ are used to show the accuracy, the computational domain is $[0, 1]$ with periodic boundary conditions. In Table 3.1, the L^1 and L^∞ error for the cell averages at time $t = 1.0$ by HWENO5-multi3 and HWENO5-RK3 methods are shown for comparison, respectively. In this example we take $\Delta t = O\big(\Delta x^{\frac{5}{3}}\big)$ for the purpose to guarantee that spatial error dominates, and we can see that both schemes achieve their designed accuracy order.

Another accuracy tests of the finite volume HWENO scheme with maximum-principle-preserving (MPP) limiter which refers to [2] for details are shown in Table 3.2. The accuracy of HWENO5-RK3 with MPP limiter degenerates clearly in very refined mesh, while the normal order was observed in HWENO5-multi3 with MPP limiter. To see how many MPP limiters were actually used in this example, we recorded the number of cells where the MPP limiter was activated. We list the average percentage of the limited cells in each stage in HWENO-RK3 and that of the limited cells in each time step in HWENO-multi3. The less percentage of HWENO-multi3 is observed, compared with that of HWENO-RK3.

Example 3.2. We solve the Burgers' equation

$$u_t + \left(\frac{u^2}{2}\right)_x = 0 \tag{3.2}$$

with the initial condition $u(x, 0) = 0.5 + \sin(\pi x)$,and a 2-periodic boundary condition. When $t = 0.5/\pi$ the solution is still smooth.

Table 3.2: HWENO5-multi3 and HWENO5-RK3 with MPP limiter, for the linear equation with initial data $u_0(x) = 0.5 + \sin(2\pi x)$. $\Delta x = \frac{1}{N}$ and $\Delta t = CFL\Delta x^{\frac{5}{3}}$. $t = 0.1$. L_1 and L_∞ error.

HWENO5-multi3 with MPP limiter							
N	L_1 error	Order	L_∞ error	Order	$u_{\min}$	u_{max}	limited (%)
20	6.47E-05		1.11E-04		-0.5	1.5	14.00
40	1.84E-06	5.14	3.58E-06	4.96	-0.5	1.5	8.55
80	5.52E-08	5.06	1.13E-07	4.98	-0.5	1.5	3.41
160	1.69E-09	5.03	3.44E-09	5.04	-0.5	1.5	1.01
320	5.26E-11	5.01	9.64E-11	5.16	-0.5	1.5	0.00
HWENO5-RK3 with MPP limiter							
N	L_1 error	Order	L_∞ error	Order	$u_{\min}$	u_{max}	limited (%)
20	5.97E-05		1.20E-04		-0.5	1.5	16.5
40	1.77E-06	5.08	4.42E-06	4.76	-0.5	1.5	8.93
80	6.81E-08	4.70	4.84E-07	3.19	-0.5	1.5	4.73
160	2.99E-09	4.51	4.92E-08	3.30	-0.5	1.5	2.43
320	1.39E-10	4.43	4.75E-09	3.37	-0.5	1.5	1.23

Table 3.3: HWENO5-multi3 and HWENO5-Rk3. Burgers' equation $u_t + (u^2/2)_x = 0$ with the initial condition $u(x, 0) = 0.5 + \sin(\pi x)$,$t = 0.5/\pi$. L_1 and L_∞ errors. Uniform meshes with N cells.

N	HWENO5-multi3				HWENO5-RK3			
	L_1 error	Order	L_∞ error	Order	L_1 error	Order	L_∞ error	Order
10	4.89E-03		1.77E-02		5.06E-03		1.79E-02	
20	4.93E-04	3.31	3.34E-03	2.41	4.93E-04	3.36	3.33E-03	2.43
40	3.65E-05	3.76	3.25E-04	3.36	3.65E-05	3.76	3.24E-04	3.36
80	1.61E-06	4.50	1.51E-05	4.43	1.61E-06	4.51	1.51E-05	4.43
160	6.25E-08	4.68	5.49E-07	4.78	6.25E-08	4.68	5.49E-07	4.78
320	1.86E-09	5.07	2.06E-08	4.74	1.86E-09	5.07	2.06E-08	4.74

In Table 3.3, the L^1 and L^∞ error for the cell averages at time $t = 0.5/\pi$ by HWENO5-multi3 and HWENO5-RK3 are displayed for comparison. We can see that the fifth order are achieved for both HWENO5-multi3 and HWENO5-RK3 schemes, and they produce similar numerical errors and orders of accuracy.

In Fig. 3.1, we give the L^1 and L^∞ numerical errors for (3.2) and CPU time when space points numbers are $N = 20, 40, 80, 160, 320$. We can see that the efficiency of HWENO5-multi3 is similar to that of HWENO5-RK3.

Example 3.3. We solve the following nonlinear system of Euler equations

$$u_t + f(u)_x = 0 \tag{3.3}$$

with

$$u = (\rho, \rho v, E)^T, \ f(u) = \left(\rho v, \rho v^2 + p, v(E + p)\right)^T.$$

Here ρ is the density, v is the velocity, E is the total energy, p is the pressure, which is related to total energy by $E = p/(\gamma - 1) + 1/2\rho v^2$ with $\gamma = 1.4$. The initial condition is set to be $\rho(x, 0) = 1 + 0.2\sin(\pi x), v(x, 0) = 1, p(x, 0) = 1$, with a 2-periodic boundary condition. The exact solution is $\rho(x, t) = 1 + 0.2\sin(\pi(x - t)), v(x, t) = 1, p(x, t) = 1$. We compute the solution

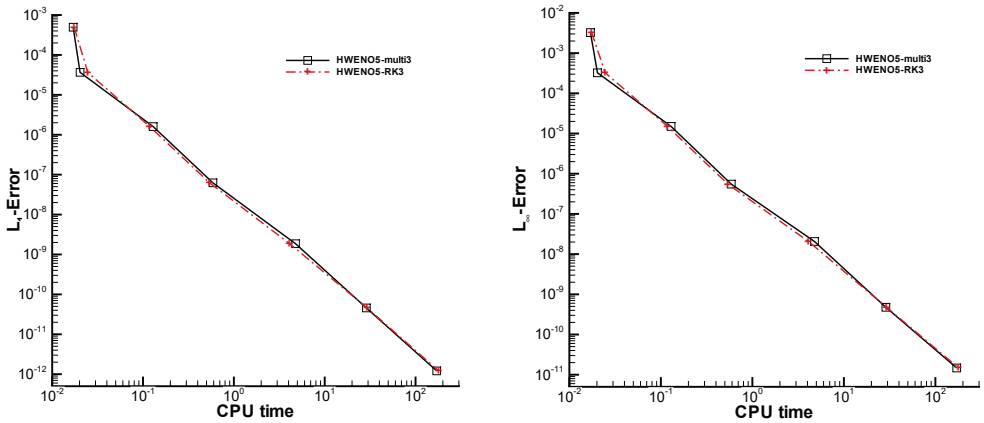

Fig. 3.1. Numerical errors and CPU time by HWENO5-multi3 and HWENO5-RK3 with $N = 20, 40, 80, 160, 320$ for the 1D Burgers' equation. Left: L^1 errors. Right: L^∞ errors.

up to $t = 2$. The errors and numerical orders of accuracy of the density ρ for the HWENO5-multi3 and HWENO5-RK3 schemes are shown in Table 3.4. We can see that both scheme achieve their designed order of accuracy.

Table 3.4: HWENO5-multi3 and HWENO5-RK3 using N equally spaced cells. Euler equations. $\rho(x, 0) = 1 + 0.2\sin(\pi x), v(x, 0) = 1, p(x, 0) = 1$. $t = 2$. L_1 and L_∞ errors of density ρ.

N	HWENO5-multi3				HWENO5-RK3			
	L_1 error	Order	L_∞ error	Order	L_1 error	Order	L_∞ error	Order
10	3.12E-03		4.94E-03		3.13E-03		4.94E-03	
20	1.17E-04	4.73	2.27E-04	4.44	1.17E-04	4.73	2.27E-04	4.44
40	3.47E-06	5.08	6.68E-06	5.08	3.47E-06	5.08	6.69E-06	5.08
80	1.07E-07	5.02	2.11E-07	4.98	1.07E-07	5.02	2.11E-07	4.98
160	3.28E-09	5.02	5.94E-09	5.15	3.29E-09	5.02	5.94E-09	5.15
320	9.83E-11	5.06	1.68E-10	5.14	9.84E-11	5.06	1.69E-10	5.14

In Fig. 3.2, we give the L^1 and L^∞ numerical errors for (3.3) and CPU time when space points numbers are $N = 10, 20, 40, 80, 160, 320$. We can see that the efficiency of HWENO5-multi3 is similar to that of HWENO5-RK3.

Example 3.4. We solve the same nonlinear Burgers equation (3.2) as in Example 3.2 with the same initial condition $u(x, 0) = 0.5 + \sin(\pi x)$, except that we now plot the results at $t = \frac{1.5}{\pi}$ when a shock has already appeared in the solution. In Fig. 3.3, the solutions of HWENO5-multi3 and HWENO-RK3 with N=80 cells are shown. The solid line is the exact solution. We can see that both schemes give nonoscillatory shock transitions for this problem.

Example 3.5. We solve the nonlinear non-convex scalar Buckley-Leverett problem

$$u_t + \left(\frac{4u^2}{4u^2 + (1-u)^2}\right)_x = 0 \tag{3.4}$$

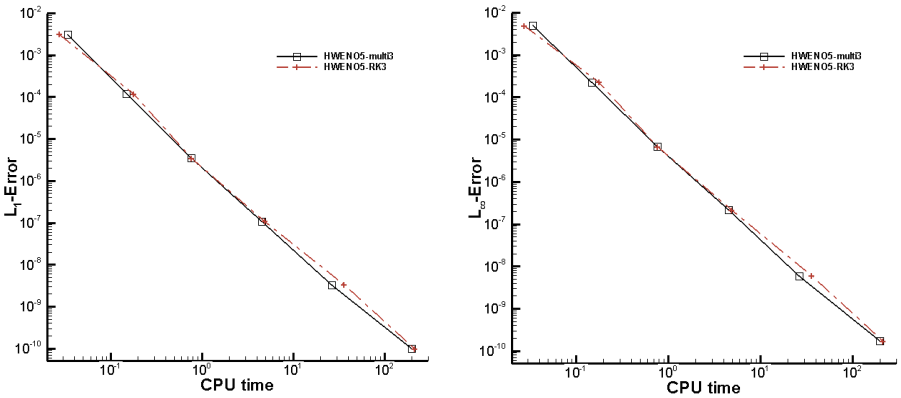

Fig. 3.2. Numerical errors and CPU time by HWENO5-multi3 and HWENO5-RK3 with $N = 10, 20, 40, 80, 160, 320$ for the 1D Euler equations order test. Left: L^1 errors. Right: L^∞ errors.

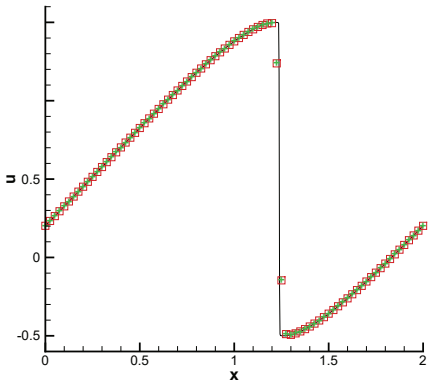

Fig. 3.3. Example 3.4 with N=80 cells. Solid line: exact solution; Squares: computed solution of HWENO5-multi3; Pluses: computed solution of HWENO5-RK3.

with the initial data $u = 1$ when $-\frac{1}{2} \leq x \leq 0$ and $u = 0$ elsewhere. The solution is computed up to $t = 0.4$. The exact solution is a shock-rarefaction-contact discontinuity mixture. In Fig. 3.4, The solutions of HWENO5-multi3 and HWENO-RK3 with $N = 80$ cells are shown. The solid line is the exact solution. We can see that both schemes give good resolution to the correct entropy solution for this problem.

Example 3.6. We solve the Euler equations (3.3) with a Riemann initial condition for the Lax Problem

$(\rho, v, p) = (0.445, 0.698, 3.528)$ for $x \leq 0$, $\quad (\rho, v, p) = (0.5, 0, 0.571)$ for $x > 0$.

The computed density ρ is plotted at $t = 1.3$ against the exact solution. In Fig. 3.5 we plot the solutions with $N = 200$ cells by HWENO5-multi3 scheme and HWENO5-RK3 scheme. We also also show the variable time step. We can see that both schemes give good non-oscillatory shock transitions for this problem.

Example 3.7. The previous examples contain only shocks and simple smooth region solutions (almost piecewise linear), for which shock resolution is the main concern and usually a good

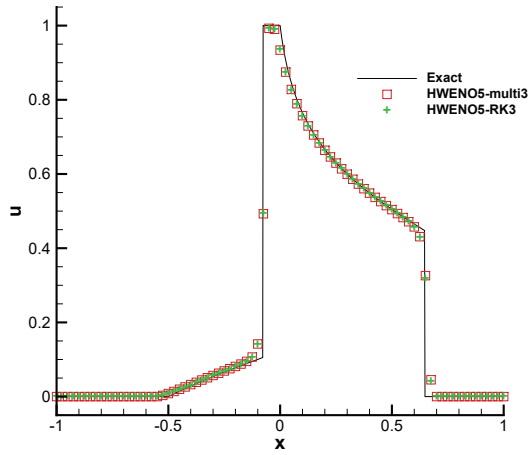

Fig. 3.4. The Buckley-Leverett problem. $t = 0.4, N = 80$ cells. Solid line: exact solution; Squares: computed solution of HWENO5-multi3; Pluses: computed solution of HWENO5-RK3.

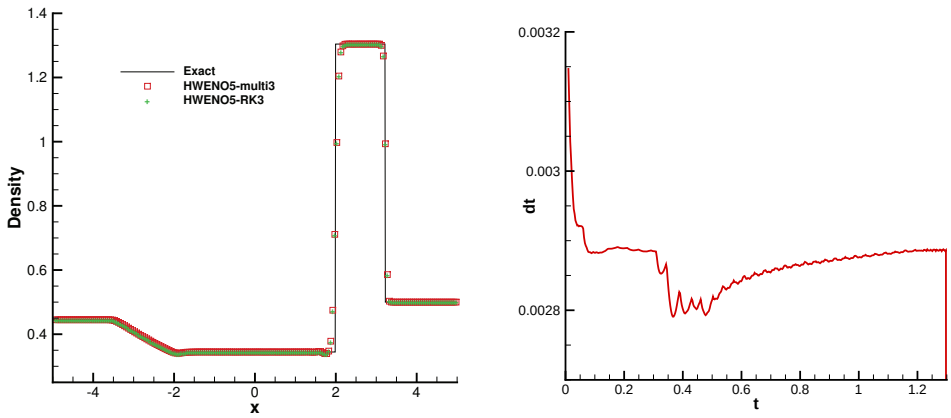

Fig. 3.5. The Lax problem. t=1.3. N=200 cells. Left: Solid line: exact solution; Squares: computed solution of HWENO5-multi3; Pluses: computed solution of HWENO5-RK3. Right: time step.

second-order non-oscillatory scheme would give satisfactory results. There is little advantage in using higher order schemes for such cases. We have been using them in the numerical experiments mainly to demonstrate the non-oscillatory properties of the high order schemes. A higher order scheme would show its advantage when the solution contains both shocks and complex smooth region structures. A typical example for this is the problem of shock interaction with entropy waves [22]. We solve the Euler equation (3.3) with a moving Mach=3 shock interacting with sine waves in density, i.e. initially

$$(\rho, v, p) = (3.857143, 2.629369, 10.333333) \text{ for } x < -4,$$

$$(\rho, v, p) = (1 + \varepsilon \sin(5x), 0, 1) \text{ for } x \geq -4.$$

Here we take $\varepsilon = 0.2$. The computed density ρ is plotted at $t = 1.8$ against the reference solution, which is a converged solution computed by the fifth-order finite difference WENO

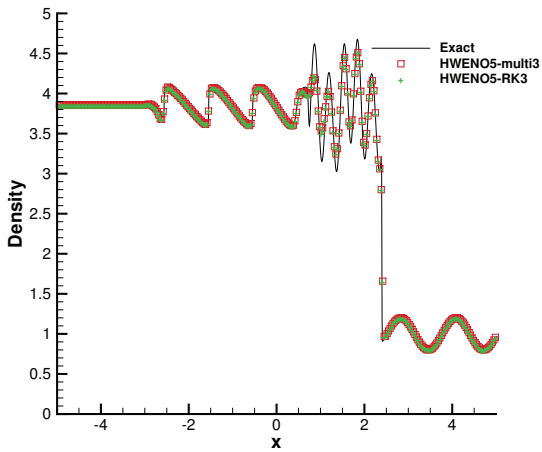

Fig. 3.6. The shock density wave interaction problem. $t = 1.8$, $N = 300$ cells. Density ρ. Solid line: "exact solution"; Squares: computed solution of HWENO5-multi3; Pluses: computed solution of HWENO5-RK3.

scheme [9] with 2000 grid points. In Fig. 3.6 we show the results of the HWENO5-multi3 scheme and the HWENO5-RK3 scheme with $N = 300$ cells. We can also see that the computational result by HWENO5-multi3 is similar to that by HWENO5-RK3.

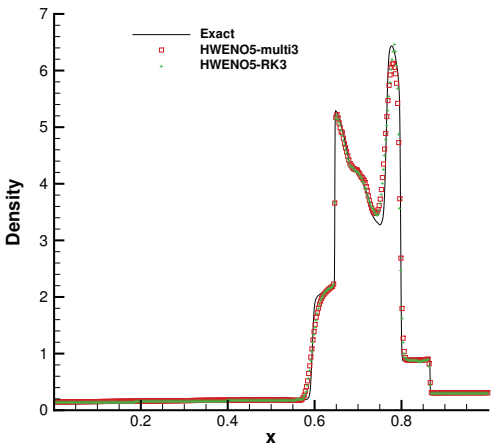

Fig. 3.7. The interaction of blast waves problem, 400 cells, $t = 0.038$, Density ρ. Solid line: "exact solution"; Squares: computed solution of HWENO5-multi3; Pluses: computed solution of HWENO5-RK3.

Example 3.8. We consider the interaction of blast waves of Euler equation (2.21) with the initial condition

$(\rho, v, p) = (1, 0, 1000)$ for $0 \le x < 0.1$,

$(\rho, v, p) = (1, 0, 0.01)$ for $0.1 \le x < 0.9$,

$(\rho, v, p) = (1, 0, 100)$ for $0.9 \le x$.

Table 3.5: Example 3.9 with $T = 0.5/\pi$. $L^1$and L^∞ errors.

N	HWENO4-multi3				HWENO4-RK3			
	L_1 error	Order	L_∞ error	Order	L_1 error	Order	L_∞ error	Order
10×10	4.98E-03		1.42E-02		5.10E-03		1.44E-02	
20×20	8.46E-04	2.56	3.66E-03	1.96	8.53E-04	2.58	3.67E-03	1.97
40×40	5.98E-05	3.82	4.39E-04	3.06	5.97E-05	3.84	4.36E-04	3.07
80×80	3.42E-06	4.13	3.41E-05	3.69	3.42E-06	4.13	3.45E-05	3.66
160×160	2.09E-07	4.03	3.36E-06	3.35	2.09E-07	4.03	3.38E-06	3.35
320×320	1.07E-08	4.28	2.02E-07	4.06	1.08E-08	4.28	2.02E-07	4.06

A reflecting boundary condition is applied to both ends. See [8, 24]. The computed density ρ is plotted at $t = 0.038$ against the reference "exact" solution, which is a converged solution computed by the fifth-order finite difference WENO scheme [9] with 2000 grid points. In Fig. 3.7 we show the results of the HWENO5-multi3 scheme and the HWENO5-RK3 scheme with 300 cells. We can also see that the computational result by HWENO5-multi3 is similar to that by HWENO5-RK3.

3.2. Two-dimensional test cases

Example 3.9. We solve the following nonlinear scalar Burger's equation in two dimensions:

$$u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0 \tag{3.5}$$

with the initial condition $u(x, y, 0) = 0.5 + \sin(\pi(x + y)/2)$ and a 4-periodic boundary condition in two directions. When $t = 0.5/\pi$, the solution is still smooth. The errors and numerical orders of accuracy for the HWENO4-multi3 and HWENO5-RK3 are shown in Table 3.5. We can see that the designed order is obtained.

In Fig. 3.8, we give the L^1 and L^∞ numerical errors for (3.5) and CPU time when space points numbers are $N = 10, 20, 40, 80, 160, 320$. We can see that the efficiency of HWENO4-multi3 is similar to that of HWENO4-RK3.

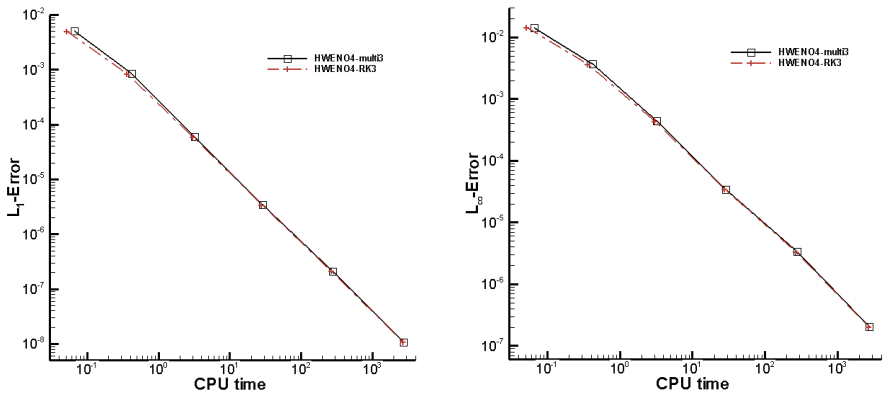

Fig. 3.8. HWENO4-multi3 and HWENO4-RK3 with $N = 10, 20, 40, 80, 160, 320$ for the 2D Burgers' equation order test. Left: L^1 errors. Right: L^∞ errors.

Table 3.6: 2D Euler equations: initial data $\rho(x,y,0) = 1+0.2\sin(\pi(x+y)), u(x,y,0) = 0.7, v(x,y,0) = 0.3, p(x,y,0) = 1$. Periodic boundary conditions in two directions. $T = 2.0$. L_1 and L_∞ errors.

cells	HWENO4-multi3				HWENO4-RK3			
	L_1 error	Order	L_∞ error	Order	L_1 error	Order	L_∞ error	Order
10× 10	1.35E-02		2.31E-02		1.35E-02		2.31E-02	
20× 20	7.29E-04	4.21	1.45E-03	4.00	7.32E-04	4.21	1.45E-03	3.99
40× 40	2.56E-05	4.83	4.99E-05	4.86	2.60E-05	4.82	5.05E-05	4.84
80× 80	1.02E-06	4.65	1.78E-06	4.81	1.06E-06	4.61	1.86E-06	4.76
160× 160	4.91E-08	4.37	7.84E-08	4.51	5.39E-08	4.30	8.60E-08	4.44
320× 320	2.75E-09	4.16	4.41E-09	4.15	3.30E-09	4.03	5.27E-09	4.03

Example 3.10. We solve the Euler equations

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ E\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ u(E+p)\end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ v(E+p)\end{pmatrix} = 0. \tag{3.6}$$

In which ρ is density, u is x-direction velocity, v is y-direction velocity, E is total energy, p is pressure. The initial conditions are: $\rho(x,y,0) = 1+0.2\sin(\pi(x+y)), u(x,y,0) = 0.7, v(x,y,0) = 0.3, p(x,y,0) = 1$, periodic boundary conditions in two directions. We compute the density solution up to $t = 2.0$. The exact solution is $\rho(x,y,0) = 1 + 0.2\sin(\pi(x+y-t))$. In Table 3.6, we show the errors and numerical orders of accuracy by HWENO4-multi3 and HWENO4-RK3 and we also can see that the designed order is obtained.

In Fig. 3.9, we give the L^1 and L^∞ numerical errors for (3.6) and CPU time when space points numbers are $N = 10, 20, 40, 80, 160, 320$. We can see that the efficiency of HWENO4-multi3 is similar to that of HWENO4-RK3.

Example 3.11. 2D Euler equations for Riemann problem [18]. We solve the Euler equations

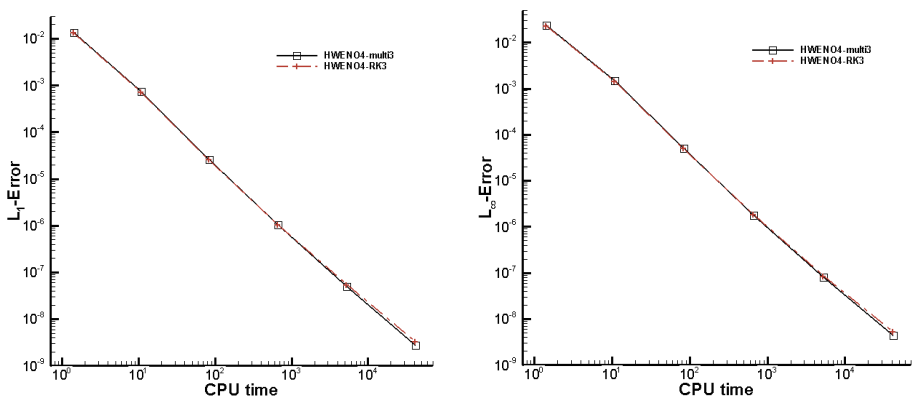

Fig. 3.9. HWENO4-multi3 and HWENO4-RK3 with $N = 10, 20, 40, 80, 160, 320$ for the two-dimensional Euler equations order test. Left: L^1 errors. Right: L^∞ errors.

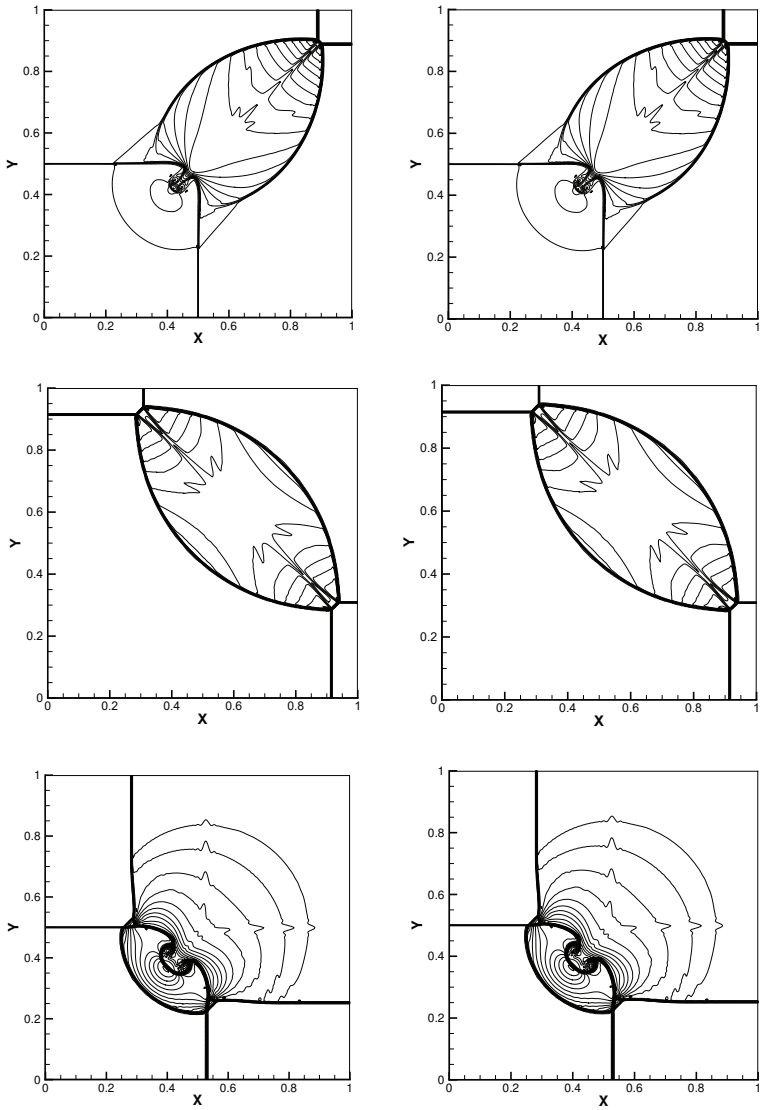

Fig. 3.10. 2D Euler equations for Riemann problem with 400×400 cells. Left: HWENO4-multi3; Right: HWENO4-RK3. From top to bottom. Initial condition (1) $T = 0.25$. 30 equally spaced density contours from 0.54 to 1.70; Initial condition (2) $T = 0.25$. 30 equally spaced density contours from 0.50 to 1.96; Initial condition (3) $T = 0.3$. 30 equally spaced density from contours 0.55 to 1.22.

(3.6) in a computational domain of $[0, 1] \times [0, 1]$ and set the initial conditions as:

$$(1) \quad (\rho, u, v, p)^T = \begin{cases} (0.5313, 0, 0, 0.4)^T, & x > 0.5, y > 0.5, \\ (1, 0.7276, 0, 1)^T, & x < 0.5, y > 0.5, \\ (0.8, 0, 0, 1)^T, & x < 0.5, y < 0.5, \\ (1, 0, 0.7276, 1)^T, & x > 0.5, y < 0.5, \end{cases}$$

$$(2) \quad (\rho, u, v, p)^T = \begin{cases} (1.1, 0, 0, 1.1)^T, & x > 0.5, y > 0.5, \\ (0.5065, 0.8939, 0, 0.35)^T, & x < 0.5, y > 0.5, \\ (1.1, 0.8939, 0.8939, 1.1)^T, & x < 0.5, y < 0.5, \\ (0.5065, 0, 0.8939, 0.35)^T, & x > 0.5, y < 0.5, \end{cases}$$

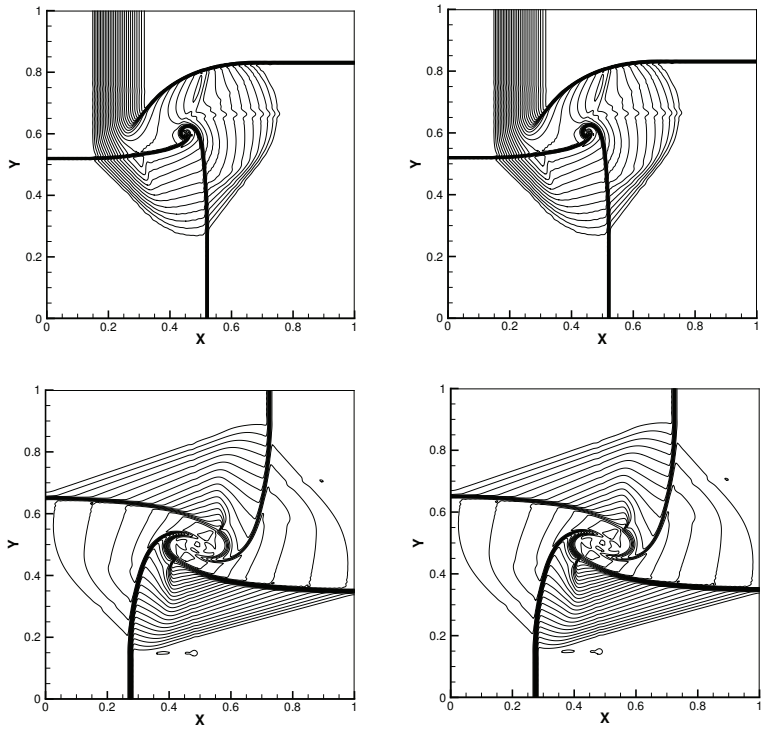

Fig. 3.11. 2D Euler equations for Riemann problem with 400×400 cells. Left: HWENO4-multi3; Right: HWENO4-RK3. From top to bottom. Initial condition (4) $T = 0.2$. 30 equally spaced density contours from 0.52 to 0.99; Initial condition (5) $T = 0.3$. 30 equally spaced density contours from 0.25 to 3.05.

$$(3)\quad (\rho, u, v, p)^T = \begin{cases} (1, 0.1, 0, 1)^T, & x > 0.5, y > 0.5, \\ (0.5313, 0.8276, 0, 0.4)^T, & x < 0.5, y > 0.5, \\ (0.8, 0.1, 0, 0.4)^T, & x < 0.5, y < 0.5, \\ (0.5313, 0.1, 0.7276, 0.4)^T, & x < 0.5, y < 0.5, \end{cases}$$

$$(4)\quad (\rho, u, v, p)^T = \begin{cases} (0.5313, 0.1, 0.1, 0.4)^T, & x > 0.5, y > 0.5, \\ (1.0222, -0.6179, 0.1, 1)^T, & x < 0.5, y > 0.5, \\ (0.8, 0.1, 0.1, 1)^T, & x < 0.5, y < 0.5, \\ (1, 0.1, 0.8276, 1)^T, & x > 0.5, y < 0.5, \end{cases}$$

$$(5)\quad (\rho, u, v, p)^T = \begin{cases} (1, 0.75, .0.5, 1)^T, & x > 0.5, y > 0.5, \\ (2, 0.75, 0.5, 1)^T, & x < 0.5, y > 0.5, \\ (1, -0.75, 0.5, 1)^T, & x < 0.5, y < 0.5, \\ (3, -0.75, .0.5, 1)^T, & x > 0.5, y < 0.5. \end{cases}$$

In the Figs. 3.10- 3.11 the computational densities by HWENO4-multi3; Right: HWENO4-RK3 schemes for the Riemann problems are shown. We can observe that most of the flow features are captured well for all these Riemann problems.

Example 3.12. Double mach reflection problem. We solve the Euler equations (3.6) in a computational domain of $[0, 4] \times [0, 1]$. A reflection wall lies at the bottom of the domain starting from $x = \frac{1}{6}, y = 0$, making a $60°$ angle with the x-axis. The reflection boundary

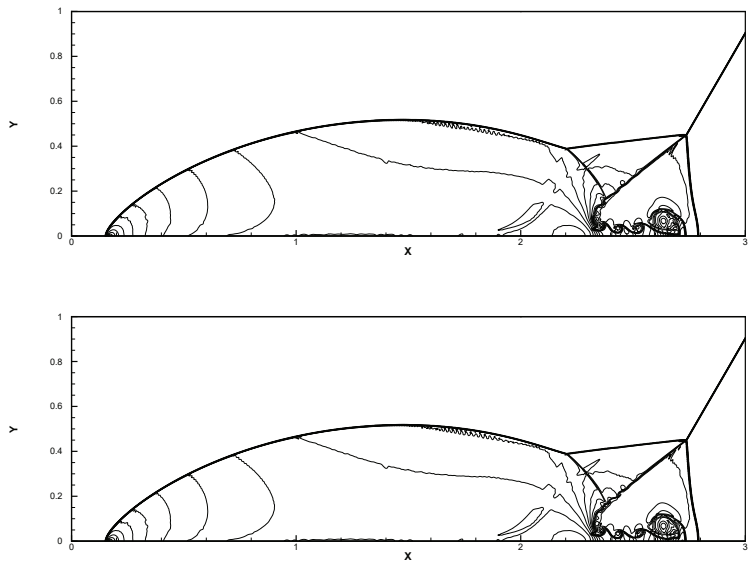

Fig. 3.12. Double Mach refection problem, 30 equally spaced density contours from 1.5 to 22.7, 2400×600.HWENO4-multi3 (Top), HWENO4-RK3 (Bottom).

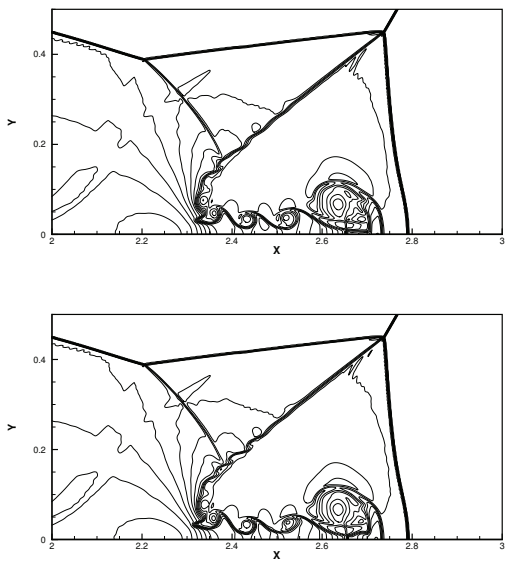

Fig. 3.13. Zoom-in pictures around the Mach stem for Double Mach refection problem, 30 equally spaced density contours from 1.5 to 22.7, 2400×600, HWENO4-multi3 (Left) and HWENO4-RK3 (Right).

condition is used at the wall, which for the rest of the bottom boundary (the part from $x = 0$ to $x = 6$), the exact post-shock condition is imposed. At the top boundary is the exact motion of the mach 10 shock. The results shown of HWENO4-multi3 with $h = \frac{1}{200}$ and HWENO4-RK3 with $h = \frac{1}{200}$ [26] are at $t = 0.2$. We present both the pictures of region $[0, 3] \times [0, 1]$ and

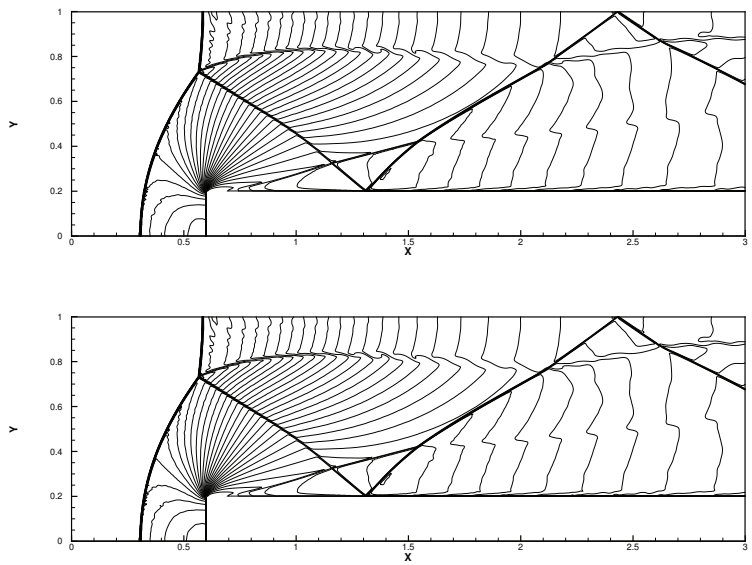

Fig. 3.14. A mach 3 wind tunnel with a step problem, 30 equally spaced density contours from 0.32 to 6.15, 1200×400 cells. HWENO4-multi3 (Top), HWENO4-RK3 (Bottom).

the blow-up region around the double mach stems in Figs. 3.12 and 3.13 respectively. All the pictures are the density contours with 30 equal spaced contour lines from 1.5 to 22.7. It is clear that HWENO4-multi3 with $h = \frac{1}{200}$ has qualitatively the similar resolution as HWENO4-RK3 with $h = \frac{1}{200}$ for the fine details of the complicated structure in this blown-up region.

Example 3.13. A mach 3 wind tunnel with a step. The setup of the problem is as follows: the wind tunnel is 1 length unit wide and 3 length units long. The step is 0.2 length units high and is located 0.6 length units from the left end of the tunnel. Initially, a right going mach 3 flow is used. Reflective boundary conditions are applied alone the walls of the tunnel and in flow and out flow boundary conditions are applied at the entrance and the exit. The computational densities by HWENO4-multi3 and HWENO4-RK3 schemes are plotted at $t = 4.0$ in Fig. 3.14 with 30 equally spaced density contours from 0.32 to 6.3. We can observe that most of the flow features are captured well by both HWENO4-multi3 and HWENO4-RK3 schemes.

4. Concluding Remarks

In this paper, we construct a class of variable step size multi-step type SSP high order temporal discretization, and we simulate the one-dimensional and two-dimensional hyperbolic conservation laws by high resolution finite volume HWENO schemes with SSP multi-step methods. We compare the numerical results with those obtained by finite volume HWENO schemes using the third-order SSP Runge-Kutta temporal discretization by addressing CPU time, efficiency and resolution. We can see that both methods have similar CPU time, efficiency, resolution and non spurious oscillatory numerical solutions.

Acknowledgments. The research is partly supported by NSFC grants 11372005, 11571290 and 91530107.

References

[1] D.S. Balsara, and C.-W. Shu. Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. *Journal of Computational Physics*, **160** (2000), 405-452.

[2] X. Cai. X. Zhang and J. Qiu. Positivity-preserving high order finite volume HWENO schemes for compressible Euler equations. *Journal of Scientific Computing*, **68** (2016), 464-483.

[3] G. Capdeville. A Hermite upwind WENO scheme for solving hyperbolic conservation laws. *Journal of Computational Physics*, **227**(2008), 2430-2454.

[4] O. Friedrich. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. *Journal of Computational Physics*, **144** (1998),194-212.

[5] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge-Kutta schemes, *Mathematics of Computation*, **67** (1998), 73-85.

[6] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability preserving high-order temporal discretization methods, *SIAM Review*, **43** (2001), 89-112.

[7] C. Hu, C.-W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, *Journal of Computational Physics*, **150** (1999), 97-127.

[8] A. Harten, B. Engquist, S. Osher, S. Chakravathy, Uniformly high order accurate essentially non-oscillatory schemes, III *Journal of Computational Physics*, **71** (1987), 231-303.

[9] G. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes *Journal of Computational Physics*, **126** (1996), 202-228.

[10] S.K. Lele, Compact finite-difference schemes with spectral-like resolution, *Journal of Computational Physics*, **103** (1992) 16-42.

[11] D. Levy, G. Puppo, G. Russo, Central WENO schemes for hyperbolic systems of conservation laws, *Mathematical Modelling and Numerical Analysis*, **33** (1999) 547-571.

[12] X.-D. Liu , S. Osher, Chan T. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, **115** (1994), 200-212.

[13] J. Qiu, C.-W. Shu, On the construction, comparison, local characteristic decomposition for high order central WENO schemes, *Journal of Computational Physics*, **183** (2002) 187-209.

[14] J. Qiu, C.W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: one dimensional case, *Journal of Computational Physics*, **193** (2004) 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, *Computers & Fluids*, **34** (2005) 642-663.

[16] J. Qiu and C.-W. Shu, Hermite WENO schemes for Hamilton-Jacobi equations. *Journal of Computational Physics*, **204** (2005), 82-99.

[17] J. Qiu. Hermite WENO Schemes with Lax-Wendroff Type temporal discretizations for Hamilton-Jacobi equations, *Journal of Computational Mathematics*, **25** (2007), 131-144.

[18] C. W. Schulz-Rinne, J. P. Collins, H. M. Glaz. Numerical solution of the Riemann problem for two-dimensional gas dynamics. *SIAM Journal on Scientific Computing*, **14**:6 (1993), 1394-1414.

[19] J. Shi, C. Hu, C.-W. Shu, A technique of treating negative weights in WENO schemes, *Journal of Computational Physics*, **175** (2002) 108-127.

[20] C.-W. Shu, Total-Variation-Diminishing temporal discretizations, *SIAM Journal on Scientific and Statistical Computing*, **9** (1988), 1073-1084.

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

[22] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes II, *Journal of Computational Physics*, **83** (1989), 32-78.

[23] B. Van-Leer, Towards the ultimate conservative difference scheme: III. A new approach to numerical convection, *Journal of Computational Physics*, **23** (1977), 276-299.

[24] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong

shocks *Journal of Computational Physics*, **54** (1984), 115-173.

[25] X. Zhang and C.-W. Shu, On maximum-principle-satisfying high order schemes for scalar conservation laws, *Journal of Computational Physics*, **229** (2010), 3091-3120.

[26] J. Zhu and J. Qiu, A Class of Forth order Finite Volume Hermite Weighted Essentially Non-oscillatory Schemes, *Science in China, Series A–Mathematics*, **51** (2008), 1549-1560.

[27] J. Zhu and J. Qiu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method III: Unstructured meshes, *Journal of Scientific Computing*, **39** (2009), 293-321.

[28] J. Zhu and J. Qiu, Hermite WENO schemes for Hamilton-Jacobi equations on unstructured meshes, *Journal of Computational Physics*, **254** (2013), 76-92.

[29] J. Zhu and J. Qiu, Finite volume Hermite WENO schemes for solving the Hamilton-Jacobi equation, *Communications in Computational Physics*, **15** (2014), 959-980.