A Numerical Comparison of the HWENO Method Based on Different Numerical Fluxes [*][†]

Hongxia Liu[‡] and Yiming Liu[§] and Juan He[¶]

Abstract: The HWENO (Hermite weighted essentially non-oscillatory) schemes are high order, high-resolution methods suitable for conservation law and convection dominated simulations with possible discontinuous or sharp gradient solutions. In most of the literature, although there are many other numerical fluxes available, the Lax-Friedrichs numerical flux is used frequently due to its simplicity. In this paper, we will study an alternative finite difference HWENO method. The core of this method is that its numerical flux framework breaks the limitations of the traditional mathematical form of numerical flux and is suitable for many different forms of numerical flux. And we systematically investigate the performance of the HWENO method and present quantitative comparisons for hyperbolic conservation laws based on different numerical fluxes. The spatial terms are discretized by using finite difference HWENO scheme and the time terms are performed by using TVD Runge-Kutta method. The HWENO method is proposed based on the original WENO methodology for solving hyperbolic conservation laws. Therefore, the HWENO scheme is similar to the classic WENO scheme achieved by using numerical flux as a building block, and their performances are closely related to the properties of the numerical fluxes. Hence, we study the performance of HWENO method based on different numerical fluxes, including the first-order monotone fluxes and second-order TVD fluxes, with the objective of obtaining better performance for the conservation laws by choosing suitable numerical fluxes. The detailed discussion focuses on the one-dimensional system case, including the issues of CPU cost, accuracy, non-oscillatory property, and resolution of discontinuities where as numerical tests are also performed for two-dimensional systems.

---

[*] The research of this author is supported by NSFC grant 11601364.

[†] The research of this author is supported by China Scholarship Council grant 201806935040.

[‡] College of Mathematics, Taiyuan University of Technology, Taiyuan, Shanxi 030024, China. E-mail: hx_ryu@163.com

[§] College of Mathematics, Taiyuan University of Technology, Taiyuan, Shanxi 030024, China. E-mail: ermao8@foxmail.com

[¶] Accounting School, Guizhou University of Finance and Economics, Guiyang, Guizhou 550025, China. E-mail: jhe@mail.gufe.edu.cn

Key words: HWENO scheme, Numerical flux, Hyperbolic conservation laws, High order accuracy

1 Introduction

By utilizing a novel form of the numerical fluxes and stencils of the interpolation polynomials, a new Hermite interpolation method is devised, known as HWENO methods, based on the original WENO [17, 19, 3, 7] scheme. In 2004 [14], Qiu and Shu first developed the finite volume HWENO scheme which based on the classical WENO philosophy and applied as limiters for the Runge-Kutta discontinuous Galerkin (RKDG) methods[14, 15]. In this work, we would like to explore an alternative flux formulation for constructing numerical fluxes which originally designed in the paper [18]. In summary, the format of this article has the following several benefits. The first advantage is the HWENO scheme inherits the benefits of the WENO scheme, namely achieving the high-order accuracy in smooth regions while maintaining stable, non-oscillatory and sharp discontinuity transitions. An additional important advantage of the HWENO scheme is its compact performance [12]. The HWENO schemes can get the same accurate order by using fewer points than the original WENO schemes. For example, five points are needed in the stencil for a fifth-order WENO (WENO5) reconstruction, while only three points are needed for a fifth-order HWENO (H-WENO5) reconstruction. The third advantage is the forms of numerical fluxes we used in the paper, which were from the Taylor series expansion. This formula of numerical fluxes breaks the limit of the mathematical form of the numerical flux. We use the point values and the corresponding derivatives to interpolate the computational cells at the interfaces, while the traditional practice of reconstructing flux functions can be applied only to smooth flux splitting to get a high-order accurate scheme. The splitting technology is very demanding in the mathematical form of numerical flux.

The HWENO method has been developed in recent years as a class of high order method for solving the conservation laws. It is widely used in many fields, such as nonlinear computational acoustic, limiters of the DG method [2], etc. In a previous work [10], we developed a class of HWENO methods which were based on the classic flux-splitting method to construct the HWENO schemes for conservation laws. The goal of this paper is to explore an alternative formulation for constructing numerical fluxes of HWENO schemes which involves interpolations of the point values u_i rather than on the flux values, while studying the performance of different numerical fluxes for the conservation laws, with the objective of obtaining better performance by choosing suitable numerical fluxes. On the other hand, in [13], Qiu studied the numerical performance of the RKDG method based on different numerical fluxes for the conservation laws. We study the performance of fluxes with the HWENO schemes.

An important component of the HWENO methods for solving the conservation laws is the numerical flux based on the exact or approximate Riemann solvers. In most of the WENO/HWENO papers, the Lax-Friedrichs (LF) numerical flux is used due to its simplicity. However, the numerical viscosity of the LF flux is almost the largest among the monotone fluxes for scalar problems. It will smooth out some useful information of numerical solutions as time progresses. There are many other numerical fluxes available in the literature. Hence, we systematically investigate the performance of the HWENO method based on different numerical fluxes, including the first-order monotone fluxes (such as the Godunov flux [20], the Engquist-Osher flux, etc.), second-order TVD fluxes and generalized Riemann solver [1, 20] based on the HWENO finite difference methods, with the objective of obtaining better performance by choosing suitable numerical fluxes. In detail, the Godunov flux [5], the Engquist-Osher (EO) flux [4, 11] for the scalar case and its extension to the system (often referred to as the Osher-Solomon flux [11]), the HLL flux [6] and its modification of HLL flux often referred to as the HLLC flux [26] are based on the approximate Riemann solver. These fluxes are two-point, first-order monotone fluxes. One of the essential two points TVD flux is the flux limiter centered (FLIC) flux [25] with the following essential two point property: $\hat{f}(u^l, u, u, u^r) = f(u)$ for any u^l and u^r, which combines a low-order monotone flux and a high-order flux with a flux limiter to guarantee the TVD-property. The other fluxes such as generalized Riemann solvers [1, 20] can also be used as a numerical flux for HWENO methods.

The present paper is organized as follows. In section 2, we give the formulations of the governing equations in the framework of the finite difference HWENO method and the updating conservative formula for conservation laws. At the same time, we give and describe the numerical fluxes considered in this paper. The numerical test is performed for the one dimensional and two-dimensional hyperbolic conservation laws in Section 3, and concluding remarks are given in Section 4.

2 Construction of the HWENO scheme

In this section, we concentrate on describing the HWENO method for conservation laws and the numerical fluxes which will be studied with the HWENO finite difference method. firstly we focus on the HWENO method for the one-dimensional case.

Consider the following scalar non-linear hyperbolic conservation law:

$$u_t + f(u)_x = 0 \tag{2.1}$$

where $f(u)$ is a convex smooth function, $f'' \geq 0$. Take the derivative of the equation (2.1) with respect to x and denoting $u_x = v$, we obtain the following coupled hyperbolic system for HWENO method:

$$\left\{ \begin{array}{l} u_t + f(u)_x = 0, \quad u(x,0) = u_0(x) \\ v_t + g(u,v)_x = 0, \quad v(x,0) = v_0(x) \end{array} \right. \tag{2.2}$$

where

$$g(u,v) = f'(u)u_x = f'(u)v. \tag{2.3}$$

By (2.3), it is easy to know that the characteristic velocity $\frac{\partial g}{\partial v}$ of the derivative equation is $f'(u)$. Hence, both equations in systems (2.2) share the same characteristic velocity: this the result is used in what follows to generate a simple approximate Riemann solver discretizing (2.2).

Set the grid as $x_i = (i - \frac{1}{2})\Delta x$ where Δx is called a spatial step. Denoting the discrete cell $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, centered on x_i. The uniform cell size will be defined by $\Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$. Conservation laws (2.2) can be written as the following system of ordinary differential equations (ODE):

$$\left\{ \begin{array}{l} \frac{du_i(t)}{dt} = -\frac{1}{\Delta x}(\hat{f}_{i+1/2} - \hat{f}_{i-1/2}) \\ \frac{dv_i(t)}{dt} = -\frac{1}{\Delta y}(\hat{g}_{i+1/2} - \hat{g}_{i-1/2}) \end{array} \right. \tag{2.4}$$

where $u_i(t)$, $v_i(t)$ are the numerical approximation to the nodal value $u(x_i,t)$ and $v(x_i,t)$ of the solution to (2.2) in a uniform grid. The numerical fluxes $\hat{f}_{i+\frac{1}{2}}$ and $\hat{g}_{i+\frac{1}{2}}$ are based on exact or approximate Riemann solvers, which must be satisfied Lipschitz continuous condition of several neighboring values u_i and v_i and also consistent with the physical fluxes $f(u)$ and $g(u,v)$.

specifically, we will use an alternative approach to construct numerical fluxes for high order conservative finite difference schemes under the flux framework in [18]. The numerical fluxes $\hat{f}_{i+\frac{1}{2}}$ and $\hat{g}_{i+\frac{1}{2}}$ are designed, so that:

$$\left\{ \begin{array}{l} \frac{1}{\Delta x}(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}) = f(u)_x|_{x=x_i} + O(\Delta x^r) \\ \frac{1}{\Delta x}(\hat{g}_{i+\frac{1}{2}} - \hat{g}_{i-\frac{1}{2}}) = g(u,v)_x|_{x=x_i} + O(\Delta x^{r-1}) \end{array} \right. \tag{2.5}$$

then the conservative difference scheme (2.4) will be the r-th order approximation to equation (2.2), when the solution is smooth. We adopt the high order flux as consisting of the low order flux plus a correction, given by:

$$\left\{ \begin{array}{l} \hat{f}_{i+\frac{1}{2}} = \hat{f}^L_{i+\frac{1}{2}} + \hat{f}^H_{i+\frac{1}{2}} \\ \hat{g}_{i+\frac{1}{2}} = \hat{g}^L_{i+\frac{1}{2}} + \hat{g}^H_{i+\frac{1}{2}} \end{array} \right. \tag{2.6}$$

where $\hat{f}^H$ and $\hat{g}^H$ were produced by Taylor expansion [18]:

$$\hat{f}^H_{i+\frac{1}{2}} = a_2\Delta x^2(\frac{\partial^2 f}{\partial x^2})_{i+\frac{1}{2}} + a_4\Delta x^4(\frac{\partial^4 f}{\partial x^4})_{i+\frac{1}{2}} + \cdots + O(\Delta x^{2m+1}), \tag{2.7}$$

$$\hat{g}^H_{i+\frac{1}{2}} = a_2\Delta x^2(\frac{\partial^2 g}{\partial x^2})_{i+\frac{1}{2}} + a_4\Delta x^4(\frac{\partial^4 g}{\partial x^4})_{i+\frac{1}{2}} + \cdots + O(\Delta x^{2m}) \tag{2.8}$$

and the constants $a_2, a_4, \cdots, a_{2m-2}, \cdots$ are existence such that the numerical flux (2.6) guarantees $r = (2m+1)th$ order accuracy in (2.5), such as $a_2 = -\frac{1}{24}$, $a_4 = \frac{7}{5760}, \cdots$. Following, we will mainly discuss the low-resolution terms $\hat{f}^L$ and $\hat{g}^L$ instead of many kinds of different numerical flux which will be given in the next section. Namely, numerical flux $\hat{f}^L_{i+\frac{1}{2}} = \hat{f}^L(u^-_{i+\frac{1}{2}}, u^+_{i+\frac{1}{2}})$ and $\hat{g}^L_{i+\frac{1}{2}} = \hat{g}^L(u^-_{i+\frac{1}{2}}, u^+_{i+\frac{1}{2}}, v^-_{i+\frac{1}{2}}, v^+_{i+\frac{1}{2}})$ can be arbitrary fluxes, $u^\pm_{i+\frac{1}{2}}$ and $v^\pm_{i+\frac{1}{2}}$ are approximations to u and v at $x_{i+\frac{1}{2}}$, respectively. One of the key benefits of this alternative numerical flux formulation is the low-resolution numerical flux $\hat{f}^L_{i+\frac{1}{2}}$ and $\hat{g}^L_{i+\frac{1}{2}}$ in (2.6) which can be any monotone flux instead of one certain numerical flux, such as LF numerical flux.

We can rewrite the conservation law (2.1) by a semi-discrete form of (2.4) and adopt the flux form of (2.6). If $u^\pm_{i+\frac{1}{2}}$ and $v^\pm_{i+\frac{1}{2}}$ are the fifth order approximations to u and v at $x_{i+\frac{1}{2}}$, respectively, then the formula (2.4) with fluxes (2.7)-(2.8) is the fifth order approximation to equation (2.2)($r = 2$). In the present work, semi-discrete systems (2.4) are becoming an ODE system, hence the time integration is performed by means of a three-stage TVD Runge-Kutta scheme [18].

2.1 Description of the low resolution terms: HWENO reconstruction

In this paper, a major building block of the alternative formulation of the high order conservative finite difference scheme discussed is the following HWENO interpolation procedure. Given the nodal values $u_i = u(x_i)$ and $v_i = v(x_i)$ of a piecewise smooth function $u(x)$ and $v(x)$, we would like to find a high-order accurate approximation of $u(x)$ and $v(x)$ at half points $x_{i+\frac{1}{2}}$. The numerical flux function in Eq. (2.4) can also be evaluated by the fifth-order finite difference HWENO scheme. For completeness, the formulations of this HWENO scheme is given below. The construction of $u^\pm_{i+\frac{1}{2}}$ and $v^\pm_{i+\frac{1}{2}}$ by a fifth-order HWENO interpolation.

The formulae of the HWENO scheme can be expressed as

$$u^-_{i+\frac{1}{2}} \approx \sum_{r=0}^{2} \omega_r u^{(r)}_{i+\frac{1}{2}} \tag{2.9}$$

where $u^{(r)}_{i+\frac{1}{2}}$ is obtained by a quadratic polynomial reconstruction of $u(x_{i+\frac{1}{2}})$ (denoted by $p_r(x)$, $p_r(x_j) = u_j, r = 0, 1, 2)$ on r-th set of candidate stencils S_r. S_r is the small stencil, $s_0 =$

$\{x_{i-1}, x_i\}$, $s_1 = \{x_i, x_{i+1}\}$, $s_2 = \{x_{i-1}, x_i, x_{i+1}\}$, which cover all possible stencils including the grid point that one point upwind to $x_{i+\frac{1}{2}}$. For r, ω_r is the nonlinear weight that satisfies

$$\sum_{r=0}^{2} \omega_r = 1, \ \ \omega_r \geq 0.$$

Following the original WENO recipe [8] (HWENO format uses the same nonlinear method), ω_r is given by

$$\omega_r = \frac{\bar{\omega}_r}{\sum_{s=0}^{k-1} \bar{\omega}_r}, \ \ r = 0, 1, 2, \tag{2.10}$$

with

$$\bar{\omega}_r = \frac{\gamma_r}{(\varepsilon + \beta_r)^2},$$

where γ_r is a linear weight which guarantees that the overall scheme (2.9) is fifth-order accurate. And β_r is the smoothness indicator, defined by

$$\beta_r = \sum_{l=1}^{2} \int_{I_i} \Delta x^{2l-1} (\frac{\partial^l p_r(x)}{\partial^l x})^2 dx. \tag{2.11}$$

We take $\varepsilon = 10^{-6}$ (a small positive number) to avoid division by zero.

Especially, for the fifth-order scheme, we have

$$\gamma_0 = \frac{1}{16}, \gamma_1 = \frac{9}{16}, \gamma_2 = \frac{3}{8}.$$

And $u^{(r)}_{i+\frac{1}{2}}$, β_r $(r = 0, 1, 2)$ can be computed by

$$u^{(0)}_{i+\frac{1}{2}} = -\frac{5}{4} u_{i-1} + \frac{9}{4} u_i - \frac{3}{4} \Delta x v_{i-1}, \tag{2.12}$$

$$u^{(1)}_{i+\frac{1}{2}} = \frac{1}{4} u_i + \frac{3}{4} u_{i+1} - \frac{1}{4} \Delta x v_{i+1}, \tag{2.13}$$

$$u^{(2)}_{i+\frac{1}{2}} = -\frac{1}{8} u_{i-1} + \frac{3}{4} u_i + \frac{3}{8} u_{i+1} \tag{2.14}$$

and

$$\beta_0 = (-2u_{i-1} + 2u_i - \Delta x v_{i-1})^2 + \frac{13}{3}(-u_{i-1} + u_i - \Delta x v_{i-1})^2, \tag{2.15}$$

$$\beta_1 = (-2u_i + 2u_{i+1} - \Delta x v_{i+1})^2 + \frac{13}{3}(u_i - u_{i+1} + \Delta x v_{i+1})^2, \tag{2.16}$$

$$\beta_2 = \frac{1}{4}(-u_{i-1} + u_{i+1})^2 + \frac{13}{12}(u_{i-1} - 2u_i + u_{i+1})^2.$$

The reconstruction procedure of $u^+_{i+\frac{1}{2}}$ is symmetric to the procedure used for $u^-_{i+\frac{1}{2}}$ with respect to $x_{i+\frac{1}{2}}$ as described above.

For the reconstruction of the derivative values $v^\pm_{i+1/2}$, a key difference is that the reconstructed stencils are different. Given the nodal values of u_i and v_i, we construct Hermite type cubic reconstruction polynomials $p_r(x)$, $(r = 0, 1, 2)$ in the small stencils $s_0 = \{x_{i-1}, x_i\}$, $s_1 = \{x_i, x_{i+1}\}$, $s_2 = \{x_{i-1}, x_i, x_{i+1}\}$, respectively. And we have

$$\gamma_0' = \frac{1}{112}, \gamma_1' = \frac{15}{16}, \gamma_2' = \frac{3}{56}. \tag{2.17}$$

For the derivative equation, the smoothness indicator become

$$\beta_j = \sum_{k=2}^{3} \int_{I_i} \Delta x^{2k-1} \left(\frac{\partial^k}{\partial x^k} p_j(x)\right)^2 dx. \tag{2.18}$$

Note that the summation begins from the second derivative rather than the first, see [14] for more details. The formulae for $v^r_{i+\frac{1}{2}}$, β_r $(r = 0, 1, 2)$ become

$$v^{(0)}_{i+\frac{1}{2}} = \frac{9}{2\Delta x}(u_{i-1} - u_i) + \frac{7}{4}v_{i-1} + \frac{15}{4}v_i, \tag{2.19}$$

$$v^{(1)}_{i+\frac{1}{2}} = \frac{3}{2\Delta x}(-u_i + u_{i+1}) - \frac{1}{4}v_i - \frac{1}{4}v_{i+1}, \tag{2.20}$$

$$v^{(2)}_{i+\frac{1}{2}} = \frac{1}{8\Delta x}(u_{i-1} - u_i + 7u_{i+1}) + \frac{1}{4}v_i, \tag{2.21}$$

and

$$\begin{aligned}
\beta_0 &= \frac{13}{12}(12(u_{i-1} - u_i) + 6\Delta x(v_{i-1} + v_i))^2 + (6(u_{i-1} - u_i) + \Delta x(2v_{i-1} + 4v_i))^2, \\
\beta_1 &= \frac{13}{12}(12(u_i - u_{i+1}) + 6\Delta x(v_i + v_{i+1}))^2 + (-6(u_i - u_{i-1}) - \Delta x(4v_i + 2v_{i+1}))^2, \\
\beta_2 &= \frac{13}{12}(-3u_{i-1} + 3u_{i+1} - 6\Delta x v_i)^2 + (u_{i-1} - 2u_i + u_{i+1})^2..
\end{aligned}$$

Again the reconstruction procedure of $v^+_{i+\frac{1}{2}}$ is symmetric to the procedure used for $v^-_{i+\frac{1}{2}}$ with respect to $x_{i+\frac{1}{2}}$ as described above.

We described the reconstruction procedure of fifth order HWENO finite difference scheme in the above. For system cases, such as the Euler equations of gas dynamics, in order to avoid oscillation, both the reconstructions of $u^-_{i+1/2}$ and $v^-_{i+1/2}$ from u_i and v_i are performed in the local characteristic directions.

2.2 Construction of the high resolution terms

To get an approximation with fifth order accuracy (r = 5 in (2.5)), we can use the first two terms given by (2.7)-(2.8). The high-resolution terms of the numerical flux in (2.7)-(2.8) have at least Δx^2 in their coefficients, hence they only need lower order approximations and they are expected to contribute much less to spurious oscillations. Therefore, we approximate these remaining terms by simple central approximation or one-point upwind-biased approximation with suitable orders of accuracy, without using the more expensive HWENO procedure.

Discrete $\hat{f}^H_{i+\frac{1}{2}}$ and $\hat{g}^H_{i+\frac{1}{2}}$. Considering the stability, we split fluxes f and g into two parts $f = f^+ + f^-$ and $g = g^+ + g^-$, here

$$f^+ = \frac{1}{2}(f(u) + \alpha u),\ f^- = \frac{1}{2}(f(u) - \alpha u),$$

$$g^+ = \frac{1}{2}(g(u,v) + \alpha v),\ g^- = \frac{1}{2}(g(u,v) - \alpha v),$$

where $\alpha = \max |f'(u)|$, and we can see that

$$\frac{d}{du}f^+ \geq 0,\ \frac{d}{du}f^- \leq 0,\ \frac{\partial}{\partial v}g^+ \geq 0,\ \frac{\partial}{\partial v}g^- \leq 0.$$

Giving the point value $f_i^+ = \frac{1}{2}(f(u_i) + \alpha u_i)$, $f_i^- = \frac{1}{2}(f(u_i) - \alpha u_i)$, $g_i^+ = \frac{1}{2}(g(u_i, v_i) + \alpha v_i)$, $g_i^- = \frac{1}{2}(g(u_i, v_i) - \alpha v_i)$, let

$$\hat{f}^H_{i+\frac{1}{2}} = \hat{f}^+_{i+\frac{1}{2}} + \hat{f}^-_{i+\frac{1}{2}} = (\hat{f}_1^+)_{i+\frac{1}{2}} + (\hat{f}_1^-)_{i+\frac{1}{2}} + (\hat{f}_2^+)_{i+\frac{1}{2}} + (\hat{f}_2^-)_{i+\frac{1}{2}}$$

$$\hat{g}^H_{i+\frac{1}{2}} = \hat{g}^+_{i+\frac{1}{2}} + \hat{g}^-_{i+\frac{1}{2}} = (\hat{g}_1^+)_{i+\frac{1}{2}} + (\hat{g}_1^-)_{i+\frac{1}{2}} + (\hat{g}_2^+)_{i+\frac{1}{2}} + (\hat{g}_2^-)_{i+\frac{1}{2}}$$

then

$$a_2 \Delta x^2 (\frac{\partial^2 f}{\partial x^2})_{i+\frac{1}{2}} \approx a_2 (\hat{f}_1^+)_{i+\frac{1}{2}} + a_2 (\hat{f}_1^-)_{i+\frac{1}{2}}, \quad a_4 \Delta x^4 (\frac{\partial^4 f}{\partial x^4})_{i+\frac{1}{2}} \approx a_4 (\hat{f}_2^+)_{i+\frac{1}{2}} + a_4 (\hat{f}_2^-)_{i+\frac{1}{2}}$$

$$a_2 \Delta x^2 (\frac{\partial^2 g}{\partial x^2})_{i+\frac{1}{2}} \approx a_2 (\hat{g}_1^+)_{i+\frac{1}{2}} + a_2 (\hat{g}_1^-)_{i+\frac{1}{2}}, \quad a_4 \Delta x^4 (\frac{\partial^4 g}{\partial x^4})_{i+\frac{1}{2}} \approx a_4 (\hat{g}_2^+)_{i+\frac{1}{2}} + a_4 (\hat{g}_2^-)_{i+\frac{1}{2}}$$

where

$$\begin{aligned}
(\hat{f}_1^-)_{i+\frac{1}{2}} &= \frac{5}{4} f^+_{i-1} - f^+_i - \frac{1}{4} f^+_{i+1} + \frac{1}{2} g^+_{i-1} + g^+_{i+1},\\
(\hat{f}_1^+)_{i+\frac{1}{2}} &= \frac{5}{4} f^-_{i+2} - f^-_{i+1} - \frac{1}{4} f^-_{i} - \frac{1}{2} g^-_{i+2} - g^-_{i},\\
(\hat{f}_2^-)_{i+\frac{1}{2}} &= -12(f^+_{i-1} - 2f^+_i + f^+_{i+1}) - 6(g^+_{i-1} - g^+_{i+1}),\\
(\hat{f}_2^+)_{i+\frac{1}{2}} &= -12(f^-_{i-1} - 2f^-_i + f^-_{i+1}) - 6(g^-_{i-1} - g^-_{i+1}),\\
(\hat{g}_1^-)_{i+\frac{1}{2}} &= \frac{1}{4}(39 f^+_{i-1} - 48 f^+_i + 9 f^+_{i+1} + 21 g^+_{i-1} + 12 g^+_i - 3 g^+_{i+1}),\\
(\hat{g}_1^+)_{i+\frac{1}{2}} &= \frac{1}{4}(-9 f^-_{i-1} + 48 f^-_i - 39 f^-_{i+1} - 3 g^-_{i-1} + 12 g^-_i + 21 g^-_{i+1}),\\
(\hat{g}_2^-)_{i+\frac{1}{2}} &= 90(f^+_{i-1} - f^+_{i+1}) + 30(g^+_{i-1} + 4 g^+_i + g^+_{i+1}),\\
(\hat{g}_2^+)_{i+\frac{1}{2}} &= 90(f^-_{i-1} - f^-_{i+1}) + 30(g^-_{i-1} + 4 g^-_i + g^-_{i+1}).
\end{aligned}$$

Next, we list the numerical fluxes under consideration, and the demonstration of their performance will be given in the next subsection.

2.3 Description of fluxes for the low resolution terms

In this paper, we use the formulation (2.6) as the numerical flux for the hyperbolic law. Although this formulation is more expensive than the standard formulation, it does have several advantages. One of the most important advantages is that arbitrary monotone fluxes can be used in this framework, while the traditional practice of reconstructing flux functions can be applied only to smooth flux splitting. We will list some frequently-used in the literature numerical flux as follows as the low-resolution term of the present paper.

2.3.1 The Lax-Friedrichs (LF) flux and the local LF (LLF) flux [25, 13, 9]

The LF flux is one of the simplest and most widely used in many literatures. However, the numerical viscosity is the largest among monotone fluxes which will polish many useful messages with the time development.

The LF or the LLF flux is defined by:

$$\hat{f}^{LF}(u^-, u^+) = \frac{1}{2}\left[\left(f(u^-) + f(u^+)\right) - \alpha\left(u^+ - u^-\right)\right], \tag{2.22}$$

$$\hat{g}^{LF}(u^-, u^+; v^-, v^+) = \frac{1}{2}\left[\left(g(u^-, v^-) + g(u^+, v^+)\right) - \alpha\left(v^+ - v^-\right)\right]. \tag{2.23}$$

where for the LF flux, α is taken as an upper bound over the whole line for $|f'(u)|$ in the scalar case, or for the absolute value of eigenvalues of the Jacobian for the system case, and for the LLF flux α is taken as an upper bound between u^- and u^+. In the process of solving with HWENO scheme, the numerical flux of the derivative equation has the same value α with original equation, because they have the same characteristic velocity.

2.3.2 The Godunov flux [5, 25].

The Godunov flux is based on the exact Riemann solver, which has the smallest numerical viscosity among all monotone fluxes for the scalar case but could be very costly to evaluate in the system case, as it often lacks explicit formulas and relies on iterative procedures for its evaluation. The Godunov flux is defined as

$$\hat{f}^{G}(u^-, u^+) = f(u(0)),$$

where $u(0)$ is the solution of the local Riemann problem at $x/t = 0$ (the solution of the local Riemann problem is a function of the single variable x/t only due to self-similarity), i.e. the exact solution to the conservation law (2.2) with the initial condition:

$$u(x,0) = \begin{cases} u^- & \text{for} \quad x \le 0, \\ u^+ & \text{for} \quad x > 0. \end{cases}$$

For the scalar case, the Godunov flux can be expressed in a closed form as

$$\hat{f}^G(u^-,u^+) = \begin{cases} \min_{u^-\le u\le u^+} f(u) & \text{if} \quad u^- \le u^+, \\ \max_{u^+\le u\le u^-} f(u) & \text{if} \quad u^- > u^+. \end{cases} \tag{2.24}$$

However, for most nonlinear systems, the Godunov flux cannot be expressed in a closed form. Its evaluation would in general require an iterative procedure. Therefore, to the numerical flux $\hat{g}_{i\pm\frac{1}{2}}$ in derivative equation, we use the frozen coefficient method and the linear hyperbolic systems theory to update the v.

2.3.3 The Engquist-Osher (EO) flux and the Osher-Solomon flux [9, 13].

The Engquist-Osher (EO) flux for the scalar case and its extension to systems (often referred to as the Osher-Solomon flux) are smoother than the Godunov flux with an almost as small numerical viscosity, and have the advantage of explicit formulas for the scalar case and for some well known physical systems, such as the Euler equations of compressible gas dynamics. Since the existence of the explicit formulas, the evaluation of the EO flux is less costly than the Godunov flux.

For the scalar case the EO flux is given by:

$$\hat{f}^{EO}(u^-,u^+) = \frac{1}{2}\left(f(u^-) + f(u^+) - \int_{u^-}^{u^+} |f'(u)|du \right), \tag{2.25}$$

For the one dimensional system case, the explicit formulas for the Osher-Solomon flux for the Euler equations is given as follows [13] which will not given here.

To the numerical flux $\hat{g}_{i\pm\frac{1}{2}}$, since the hyperbolic equation and its derivative have the same characteristic velocity, linear hyperbolic systems theory was used with the frozen coefficient method.

2.3.4 The Harten-Lax-van Leer (HLL) flux [26, 25].

For the purpose of computing a Godunov flux, Harten, Lax and van Leer [6] presented a novel approach (HLL flux) for solving the Riemann problem approximately. The central idea is to assume, for the solution, a wave configuration that consist of two waves separating three constant states. The HLL flux for the Euler equations (one dimensional case) is defined by:

$$\hat{f}^{HLL}(u^-,u^+) = \begin{cases} f(u^-), & \text{if} \quad 0 \le s^-, \\ \frac{s^+ f(u^-) - s^- f(u^+) + s^- s^+ (u^+ - u^-)}{s^+ - s^-}, & \text{if} \quad s^- \le 0 \le s^+, \\ f(u^+), & \text{if} \quad s^+ \le 0. \end{cases} \tag{2.26}$$

$$\hat{g}^{HLL}(u^-,u^+;v^-,v^+) = \begin{cases} g(u^-,v^-)v^-, & \text{if} \quad 0 \le s^-, \\ \frac{s^+ g(u^-,v^-)v^- - s^- g(u^+,v^+)v^+ + s^- s^+ (v^+ - v^-)}{s^+ - s^-}, & \text{if} \quad s^- \le 0 \le s^+, \\ g(u^+,v^+)v^+, & \text{if} \quad s^+ \le 0. \end{cases} \tag{2.27}$$

where the lower and upper bounds of the wave speed, s^- and s^+, must be estimated. We use the pressure-velocity estimates

$$s^- = v^- - c^- q^-, \qquad s^* = v^*, \qquad s^+ = v^+ + c^+ q^+ \tag{2.28}$$

where, for $K = \pm$,

$$q^K = \begin{cases} 1, & \text{if} \quad p^* \le p^K, \\ (1 + \frac{\gamma+1}{2\gamma}(p^*/p^K - 1))^{1/2}, & \text{if} \quad p^* > p^K \end{cases}$$

with

$$p^* = \frac{1}{2}(p^- + p^+) - \frac{1}{2}(v^+ - v^-)\overline{\rho}\,\overline{c}, \qquad v^* = \frac{1}{2}(v^- + v^+) - \frac{p^+ - p^-}{2\overline{\rho}\,\overline{c}},$$

and

$$\overline{\rho} = \frac{1}{2}(\rho^- + \rho^+), \qquad \overline{c} = \frac{1}{2}(c^- + c^+).$$

But this flux has a serious flaw of diffusing contact surfaces. This is mainly because the HLL scheme is a assumption of a two wave configuration, therefore neglects the contact discontinuous. So a remedy version to this problem of intermediate waves in the HLL scheme was given in [26], named HLLC flux.

2.3.5 The HLLC flux-a modification of the HLL flux [23, 25].

The HLLC flux is a modification of the HLL flux, whereby the missing contact and shear waves are restored. The HLLC flux for the Euler equations is given by:

$$\hat{f}^{HLLC}(u^-, u^+) = \begin{cases} f(u^-), & \text{if} \quad 0 \le s^-, \\ f(u^-) + s^-(u^{*-} - u^-), & \text{if} \quad s^- \le 0 \le s^*, \\ f(u^+) + s^+(u^{*+} - u^+), & \text{if} \quad s^* \le 0 \le s^+, \\ f(u^+), & \text{if} \quad s^+ \le 0 \end{cases} \tag{2.29}$$

$$\hat{g}^{HLLC}(u^-, u^+; v^-, v^+) = \begin{cases} g(u^-, v^-)v^-, & \text{if} \quad 0 \le s^-, \\ g(u^-, v^-)v^- + s^-(v^{*-} - v^-), & \text{if} \quad s^- \le 0 \le s^*, \\ g(u^+, v^+)v^+ + s^+(v^{*+} - v^+), & \text{if} \quad s^* \le 0 \le s^+, \\ g(u^+, v^+)v^+, & \text{if} \quad s^+ \le 0 \end{cases} \tag{2.30}$$

where, for $K = \pm$,

$$u^{*K} = \rho^K \frac{s^K - v^K}{s^K - s^*} \begin{bmatrix} 1 \\ s^* \\ \frac{E^K}{\rho^K} + (s^* - v^K)[s^* + \frac{p^K}{\rho^K(s^K - v^K)}] \end{bmatrix} \tag{2.31}$$

$$v^{*K} = {\rho_x}^K \frac{s^K - v^K}{s^K - s^*} \begin{bmatrix} 1 \\ s^* \\ \frac{{E_x}^K}{{\rho_x}^K} + (s^* - v^K)[s^* + \frac{p^K}{{\rho_x}^K(s^K - v^K)}] \end{bmatrix} \tag{2.32}$$

The definitions of s^-, s^* and s^+ are given in (2.28).

2.3.6 The first-order centered (FORCE) flux [24].

This flux is the average of the LF flux and the second order Richtmyer flux, hence its viscosity is smaller than that of the LF flux. The FORCE flux is defined by:

$$\hat{f}^{FORCE}(u^-, u^+) = \frac{1}{2}\left(\hat{f}^{LF}(u^-, u^+) + \hat{f}^{R}(u^-, u^+)\right), \tag{2.33}$$

$$\hat{g}^{FORCE}(u^-, u^+; v^-, v^+) = \frac{1}{2}\left(\hat{g}^{LF}(v^-, v^+) + \hat{g}^{R}(v^-, v^+)\right), \tag{2.34}$$

where $\hat{f}^R$ is the second order Richtmyer flux given by

$$\hat{f}^R(u^-, u^+) = f(u^*), \qquad u^* = \frac{1}{2}\left(u^- + u^+ - \frac{\Delta t}{\Delta x}(f(u^+) - f(u^-))\right), \tag{2.35}$$

and the $\hat{g}^R$ is

$$\hat{g}^R(u^-, u^+; v^-, v^+) = g(v^*), \qquad v^* = \frac{1}{2}\left(v^- + v^+ - \frac{\Delta t}{\Delta x}(g(u^+, v^+) - g(u^-, v^-))\right). \tag{2.36}$$

2.3.7 A flux limiter centered (FLIC) flux [22].

The general flux limiter approach combines a low order monotone flux and a high order flux. The FLIC flux we use has the FORCE flux as the low order flux and the Richtmyer flux as the high order flux:

$$\hat{f}^{FLIC}(u^-, u^+) = \hat{f}^{FORCE}(u^-, u^+) + \phi_{i+1/2}[\hat{f}^R(u^-, u^+) - \hat{f}^{FORCE}(u^-, u^+)]. \tag{2.37}$$

where $\phi_{i+1/2}$ is a flux limiter. There are several possible choices for the flux limiter such as the superbee, van Leer and the minbee flux limiters. Following [25, 21], for the Euler equation we use the following procedure: we first define $q = E$ (total energy) and set

$$r^-_{i+1/2} = \frac{\Delta q_{i-1/2}}{\Delta q_{i+1/2}}, \qquad r^+_{i+1/2} = \frac{\Delta q_{i+3/2}}{\Delta q_{i+1/2}}$$

where $\Delta q_{i-1/2} = \overline{q}_i - \overline{q}_{i-1}$, and $\overline{q}_i$ is the cell average of q on the cell I_i. We then compute a single flux limiter

$$\phi_{i+1/2} = \min(\phi(r^-_{i+1/2}),\ \phi(r^+_{i+1/2}))$$

and apply it to all components of the flux. In this paper we use the minbee limiter:

$$\phi(r) = \begin{cases} 0, & r \le 0, \\ r, & 0 \le r \le 1, \\ 1, & r \ge 1. \end{cases}$$

To the equation and its derivative equation, we choose the same value of minbee limiter $\phi(r)$. Clearly, if $u^- = u^+ = u$, then $\hat{f}^{FLIC}(u,u) = f(u)$. Hence even if the FLIC flux depends on more than the two points u^- and u^+ through the limiter $\phi_{i+1/2}$ and we are abusing notations when we denote it by $\hat{f}^{FLIC}(u^-,u^+)$, it is indeed an essentially two point flux as defined before.

2.3.8 The multi-stage predictor-corrector (MUSTA) flux [24].

The MUSTA flux is a multi-stage predictor-corrector flux. Following [21] we use the FORCE flux as the predictor flux. The procedure to evaluate a L-stage MUSTA flux can be described as following: first we set $u_0^- = u^-$ and $u_0^+ = u^+$ for the initial stage $l = 0$, then we perform the following steps:

1. Compute the FORCE flux $\hat{f}_l^{FORCE} = \hat{f}^{FORCE}(u_l^-, u_l^+)$ on the data at the stage l.
2. If the desired number of total stages L has been reached (that is $l = L$), then the computation of the MUSTA flux is complete and the final flux is given by $\hat{f}^{MUSTA}(u^-,u^+) = \hat{f}_l^{FORCE}$. Otherwise, continue to compute the values for the next stage using:
$$u_{l+1}^- = u_l^- - \frac{\Delta t}{\Delta x}(\hat{f}_l^{FORCE} - f(u_l^-)), \qquad u_{l+1}^+ = u_l^+ - \frac{\Delta t}{\Delta x}(f(u_l^+) - \hat{f}_l^{FORCE})$$
and proceed back to step (a).

In this paper we use $L = 2$ as suggested in [21].

Remark 1: *For the system cases, we basically use the Roe type characteristic HWENO scheme in order to achieve better qualities. That is to say, the HWENO reconstruction is always applied with a local characteristic field decomposition procedure, see e.g. [16] for details, while the up-wind linear reconstruction is used for the high order derivative terms. The Roe type HWENO scheme is less dissipative and thus achieves higher resolution than the HWENO scheme. For the two-dimensional case, the reconstruction to fluxes is based on a dimension by dimension fashion.*

2.4 Extension to multiple dimensions

To present the extension of the alternative flux to multiple dimensions we consider the two-dimensional case

$$\begin{cases} u_t + f(u)_x + g(u)_y = 0, \ (x,y) \in R^2, \, t \in (0,\infty) \\ u(x,y,0) = u_0(x,y), \ (x,y) \in R^2, \end{cases} \tag{2.38}$$

where $f(u(x,y,t))$ and $g(u(x,y,t))$ are flux in x direction and y direction. Let $v = u_x$, $w = u_y$, taking the derivative x and y of (2.38) separately, then we obtain:

$$\begin{cases} u_t + f(u)_x + g(u)_y = 0, \\ v_t + h(u,v)_x + r(u,v)_y = 0, \\ w_t + q(u,w)_x + s(u,w)_y = 0 \end{cases} \tag{2.39}$$

where

$$h(u,v) = f'(u)v, \ r(u,v) = g'(u)v, \ \ q(u,w) = f'(u)w, \ s(u,w) = g'(u)w.$$

For simplicity, we assume that the mesh is uniform with the cell centers $(x_i, y_j) = (\frac{1}{2}(x_{i+1/2}+x_{i-1/2}), \frac{1}{2}(y_{j+1/2}-y_{j-1/2}))$ and cell size $x_{i+1/2}-x_{i-1/2} = \Delta x$, $y_{j+1/2}-y_{j-1/2} = \Delta y$. We also denote the cells by $I_{ij} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$. We can write (2.39) directly using a conservation approximation to the spatial derivatives

$$\begin{cases} \frac{du_{i,j}(t)}{dt} + \frac{1}{\Delta x}(\hat{f}_{i+\frac{1}{2},j} - \hat{f}_{i-\frac{1}{2},j}) + \frac{1}{\Delta y}(\hat{g}_{i,j+\frac{1}{2}} - \hat{g}_{i,j-\frac{1}{2}}) = 0, \\ \frac{dv_{i,j}(t)}{dt} + \frac{1}{\Delta x}(\hat{h}_{i+\frac{1}{2},j} - \hat{h}_{i-\frac{1}{2},j}) + \frac{1}{\Delta y}(\hat{r}_{i,j+\frac{1}{2}} - \hat{r}_{i,j-\frac{1}{2}}) = 0, \\ \frac{dw_{i,j}(t)}{dt} + \frac{1}{\Delta x}(\hat{q}_{i+\frac{1}{2},j} - \hat{q}_{i-\frac{1}{2},j}) + \frac{1}{\Delta y}(\hat{s}_{i,j+\frac{1}{2}} - \hat{s}_{i,j-\frac{1}{2}}) = 0. \end{cases} \tag{2.40}$$

For the two dimensional case, we adopt a dimension-by-dimension method to deal with $\hat{f}_{i\pm\frac{1}{2},j}$, $\hat{g}_{i,j\pm\frac{1}{2}}$, $\hat{h}_{i\pm\frac{1}{2},j}$ and $\hat{s}_{i,j\pm\frac{1}{2}}$, the mixed derivative terms $\hat{r}_{i,j\pm\frac{1}{2}}$ and $\hat{q}_{i,j\pm\frac{1}{2}}$ use the classical WENO scheme. One of the advantages of the HWENO scheme is the compactness, hence using the WENO scheme to reconstruction the mixed derivative terms must be confined by the node numbers of the HWENO scheme which we developed in the above for the compactness of our schemes. This lead to the order of our schemes lower one order accuracy than the WENO scheme, becoming fourth order precision.

The reconstruction of fluxes for the mixed derivative terms $\hat{q}_{i,j\pm\frac{1}{2}}$ will be developed by

$$\begin{aligned} \hat{q}_{i+\frac{1}{2},j} = \frac{1}{2}(q(u^+_{i+\frac{1}{2},j}, w^+_{i+\frac{1}{2},j}) + q(u^-_{i+\frac{1}{2},j}, w^-_{i+\frac{1}{2},j})) - \frac{\alpha}{2}(w^+_{i+\frac{1}{2},j} - w^-_{i+\frac{1}{2},j}) \\ -\frac{1}{24}\Delta x^2(\frac{\partial^2 q}{\partial x^2})_{i+\frac{1}{2},j} \end{aligned} \tag{2.41}$$

with $\alpha = \max_u |f'(u)|$. The reconstruction procedure of $u^\pm_{i+\frac{1}{2}}$ has been given in the second 2.1. We would like to reconstruct $w^\pm_{i+1/2,j}$ by third-order WENO reconstruction in order to keep the compactness of the scheme, this leads to the scheme is the fourth order. To the derivative term $\Delta x^2(\frac{\partial^2 q}{\partial x^2})_{i+\frac{1}{2},j}$ in (2.41), we use the same method with section 2.2 to discretize. The difference approximations as follows

$$\Delta x^2(\frac{\partial^2 q}{\partial x^2})_{i+\frac{1}{2},j} \approx q^+_{i-1,j} - 2q^+_{i,j} + q^+_{i+1,j} + q^-_{i,j} - 2q^-_{i+1,j} + q^-_{i+2,j}$$

with

$$q_{ij}^{+} = \frac{1}{2}(q(u_{ij}, w_{ij}) + \alpha w_{ij}),\ q_{ij}^{-} = \frac{1}{2}(q(u_{ij}, w_{ij}) - \alpha w_{ij}),$$

the α adopt the same value with section 2.2.

3 Numerical results

In the present section, extensive numerical experiments are performed to compare the performance of the HWENO scheme, such as accuracy, CPU time as well as the resolution of the different numerical fluxes outlined in the previous section. Comparisons are concentrated mainly on the one-dimensional cases firstly. Numerical tests are also performed for the two dimensional systems. In all the numerical examples, we take the CFL number is 0.2 for one dimensional and two-dimensional cases. In the numerical results figures, the solid line stands for the reference solution and "□" and "+" represent the numerical solution. In the time history figure, each symbol "□" and "+" denote the position reconstruction of flux by HWENO reconstruction.

Firstly we consider the one dimensional Euler equations by the present HWENO scheme. The PDEs of one dimensional Euler equations in the physical space with the Cartesian coordinate (x, y) are as follows

$$\begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix}_t + \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{bmatrix}_x = 0, \tag{3.42}$$

where ρ is the density, u is the velocity in an $x-$direction, E is the total energy, and p is the pressure, which is related to the total energy by $E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u^2$ with $\gamma = 1.4$.

Example 3.1. The first test problem is the effective order of accuracy of the scheme problem. We solve the one-dimensional nonlinear system of Euler equations (3.42). 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 up to $t = 2$. In Table 3.1 we provide a CPU time comparison for the HWENO schemes with different fluxes. The numerical errors and the orders of accuracy for the density ρ, and ratios of the numerical errors for comparison with the RKDG-LF scheme are shown in Table 3.2.

We can see that the HWENO-LF scheme costs the least CPU time among all the numerical fluxes, but at the same, it has the largest numerical errors.

From the CPU time aspect, the HWENO-G and HWENO-EO schemes cost about twice that of the HWENO-LF scheme, the HWENO-MUSTA schemes cost about 30%-40% more

Table 3.1: CPU time (in seconds) for the HWENO methods with different fluxes, for the accuracy test problem. Total CPU time for $N = 10, 20, 40, 80, 160$ and 320 cells is recorded.

flux	time	flux	time	flux	time	flux	time	flux	time
LF	61.7051	EO	112.9668	HLLC	67.0252	FLIC	67.2668	LLF	61.9344
G	98.6506	HLL	62.9382	FORCE	66.5397	MUSTA	87.8203	–	–

than that of the HWENO-LF scheme, the HWENO-FORCE, HWENO-FLIC and HWENO-HLLC schemes cost about 10% than that of the HWENO-LF scheme, and the HWENO-HLL and HWENO-LLF schemes cost about the same as that of the HWENO-LF scheme. Of course, this CPU time comparison depends on our specific implementation of these fluxes and also on the specific test case (for the Godunov flux which has an iteration procedure and may converge with a different number of steps for different solutions), but it does give the correct ball-park of the relative CPU costs of the HWENO method using these different fluxes.

From the numerical errors aspect, all other schemes for the same meshes are about 40% and 50% of that by the HWENO-LF scheme, except for the RKDG-FLIC and RKDG-FORCE schemes, which have errors about 10%-20% of that by the HWENO-LF scheme. This indicates that we have to be cautious when discussing the accuracy advantage of various fluxes as this may depend on the order of accuracy of the scheme.

We can also see that all schemes achieve their designed orders of accuracy, as expected.

Example 3.2. The second test case is the Riemann problem proposed by Lax:

$$(\rho, v, p) = \begin{cases} (0.445, 0.698, 3.528) & \text{if } x \leq 0, \\ (0.5, 0, 0.571) & \text{if } x > 0, \end{cases}$$

In this example, we test the previous numerical example to show the spurious oscillations. The computational domain is $[-5, 5]$ with 200 cells, and the final time is $t = 1.3$. In Figs. 3.1, the computed densities ρ are plotted against the reference solution and against the numerical solution computed by the HWENO-LF scheme on the same mesh, the zoomed figures in the region $[-1, 4]$ is shown which contains the contact discontinuity and the shock.

From the results, we can see that the HWENO-G, HWENO-EO, HWENO-HLL, and HWENO-HLLC schemes are slightly better than that computed by the HWENO-LF scheme, in terms of the resolution of the discontinuities, and the results computed by all other schemes are similar to that computed by the HWENO-LF scheme.

In Fig. 3.2, we compare our scheme of this article with the classic HWENO-LF method [10](fifth order accurate). Both algorithms use the same Lax-Friedrichs numerical flux. According to the results of these comparisons are demonstrated in this example, we can observe that a similar results is obtained by using our scheme.

Table 3.2: Euler equation, $\rho(x,0) = 1 + 0.2sin(\pi x)$, $v(x,0) = 1$, $p(x,0) = 1$, using N equally spaced cells with different fluxes,$t = 2$, L_1 and L_∞ errors of density ρ.

N	Flux	L_1 error	L_1 order	error ratio	L_∞ error	L_∞ order	Error ratio
10	LF	5.42E-03		1.0000	7.82E-03		1.0000
	G	2.72E-03		0.5018	3.86E-03		0.4936
	EO	2.74E-03		0.5055	3.84E-03		0.4910
	HLL	3.06E-03		0.5646	4.25E-03		0.5434
	HLLC	2.73E-03		0.5037	3.89E-03		0.4974
	FORCE	1.92E-03		0.3542	2.99E-03		0.3824
	FLIC	1.81E-03		0.3339	2.70E-03		0.3453
	MUSTA	3.38E-03		0.6236	4.42E-03		0.5652
20	LF	2.37E-04	4.52	1.0000	3.71E-04	4.40	1.0000
	G	9.98E-05	4.77	0.4211	1.77E-04	4.44	0.4771
	EO	1.01E-04	4.76	0.4262	1.78E-04	4.43	0.4798
	HLL	1.17E-04	4.71	0.4937	2.11E-04	4.33	0.5687
	HLLC	1.00E-04	4.77	0.4219	1.78E-04	4.45	0.4798
	FORCE	5.86E-05	5.03	0.2473	1.13E-04	4.73	0.3046
	FLIC	3.86E-05	5.55	0.1629	9.56E-05	4.82	0.2577
	MUSTA	1.34E-04	4.66	0.5654	2.27E-04	4.28	0.6119
40	LF	7.11E-06	5.06	1.0000	1.31E-05	4.82	1.0000
	G	2.95E-06	5.08	0.4149	5.48E-06	5.02	0.4183
	EO	2.98E-06	5.08	0.4191	5.52E-06	5.01	0.4214
	HLL	3.48E-06	5.07	0.4895	6.87E-06	4.94	0.5244
	HLLC	2.95E-06	5.08	0.4149	5.48E-06	5.02	0.4183
	FORCE	1.69E-06	5.12	0.2377	3.44E-06	5.03	0.2626
	FLIC	6.72E-07	5.84	0.0945	2.57E-06	5.21	0.1962
	MUSTA	3.99E-06	5.07	0.5612	7.39E-06	4.94	0.5641
80	LF	2.18E-07	5.03	1.0000	4.09E-07	5.00	1.0000
	G	9.08E-08	5.02	0.4165	1.71E-07	5.00	0.4181
	EO	9.18E-08	5.02	0.4211	1.73E-07	5.00	0.4230
	HLL	1.07E-07	5.02	0.4908	2.18E-07	4.98	0.5330
	HLLC	9.08E-08	5.02	0.4165	1.71E-07	5.00	0.4181
	FORCE	5.17E-08	5.03	0.2372	9.99E-08	5.11	0.2443
	FLIC	1.61E-08	5.38	0.0739	8.40E-08	4.94	0.2054
	MUSTA	1.23E-07	5.02	0.5642	2.30E-07	5.00	0.5623
160	LF	6.69E-09	5.02	1.0000	1.22E-08	5.07	1.0000
	G	2.80E-09	5.02	0.4185	5.01E-09	5.09	0.4107
	EO	2.83E-09	5.02	0.4230	5.06E-09	5.09	0.4148
	HLL	3.30E-09	5.02	0.4933	6.47E-09	5.07	0.5303
	HLLC	2.80E-09	5.02	0.4185	5.01E-09	5.09	0.4107
	FORCE	1.59E-09	5.02	0.2377	2.84E-09	5.14	0.2328
	FLIC	4.59E-10	5.13	0.0686	2.65E-09	4.99	0.2172
	MUSTA	3.78E-09	5.02	0.5650	6.78E-09	5.09	0.5557
320	LF	2.00E-10	5.06	1.0000	3.42E-10	5.16	1.0000
	G	8.41E-11	5.06	0.4205	1.43E-10	5.13	0.4181
	EO	8.50E-11	5.06	0.4250	1.44E-10	5.13	0.4211
	HLL	9.89E-11	5.06	0.4945	1.82E-10	5.15	0.5322
	HLLC	8.41E-11	5.06	0.4205	1.43E-10	5.13	0.4181
	FORCE	4.80E-11	5.05	0.2400	8.11E-11	5.13	0.2371
	FLIC	1.38E-11	5.06	0.0690	9.17E-11	4.85	0.2681
	MUSTA	1.13E-10	5.06	0.5650	1.93E-10	5.14	0.5643

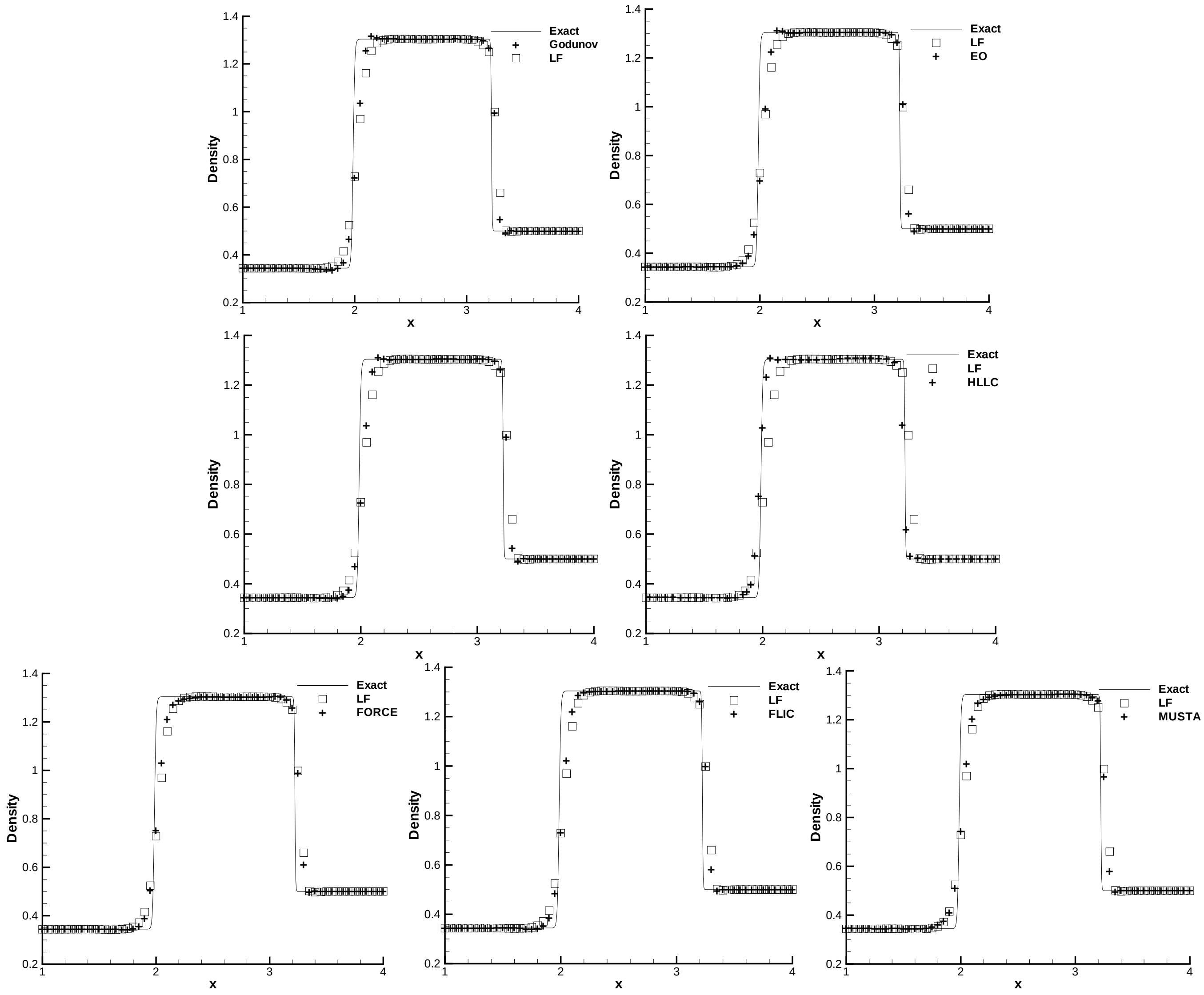

Figure 3.1: Lax problem. $t = 1.3$. HWENO with different fluxes, 200 cells. Density. Solid lines: the exact solution; hollow squares: the results computed by the HWENO-LF scheme; plus symbols: results computed by the HWENO-G (top left), HWENO-EO (top right), HWENO-HLL (middle left), HWENO-HLLC (middle right), HWENO-FORCE (bottom left), HWENO-FLIC (bottom middle) and HWENO-MUSTA (bottom right) schemes.

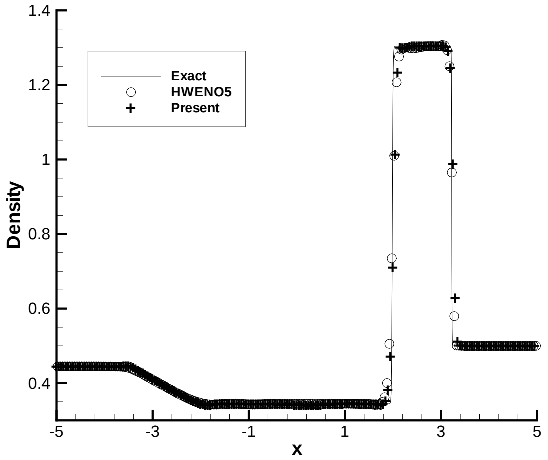

Figure 3.2: The Euler equations Lax problem. Solid line: exact solution; circle: computed solution of classical HWENO5 scheme with LF flux, plus: computed solution of present scheme.

Example 3.3. We solve the Euler equations (3.42) with a moving Mach=3 shock interacting with sine waves in density, i.e. the famous Shu-Osher problem which solution contains both shocks and complex smooth region structures. This model problem would show the advantage of different high order schemes in terms of the high resolutions with the initial condition

$$(\rho, v, p) = \begin{cases} (3.857143, 2.629369, 10.333333) & \text{if } x < -4, \\ (1 + \varepsilon \sin 5x, 0, 1) & \text{if } x \geq -4. \end{cases}$$

Here we take the $\varepsilon = 0.2$. The computed density ρ is plotted at $t = 1.8$ against the reference solution, which is a converged solution, computed using a fifth-order WENO scheme [8] by 2000 grid points, and against the solution computed by the HWENO-LF scheme on the same mesh. The computational domain is $[-5, 5]$ with 300 grids. In Figs. 3.3, "zoomed-in" pictures at the region $[0.5, 2.5]$ show the results which contains the complicated wave pattern in the smooth part of the solution.

Comparing the results from the pictures, we can observe that an improvement of resolution for the complicated wave pattern in this example for all the other schemes over the HWENO-LF scheme. All the other schemes have similar performance for this example.

Similar to the previous example, we show a comparison of the numerical results by our scheme and standard HWENO-LF discretization scheme in Fig. 3.4. we can see that the results by a method with alternative HWENO-LF(our scheme) has better performance and resolution than the classic HWENO-LF scheme.

Example 3.4. We solve the interaction of blast waves of the Euler equations (3.42) for a

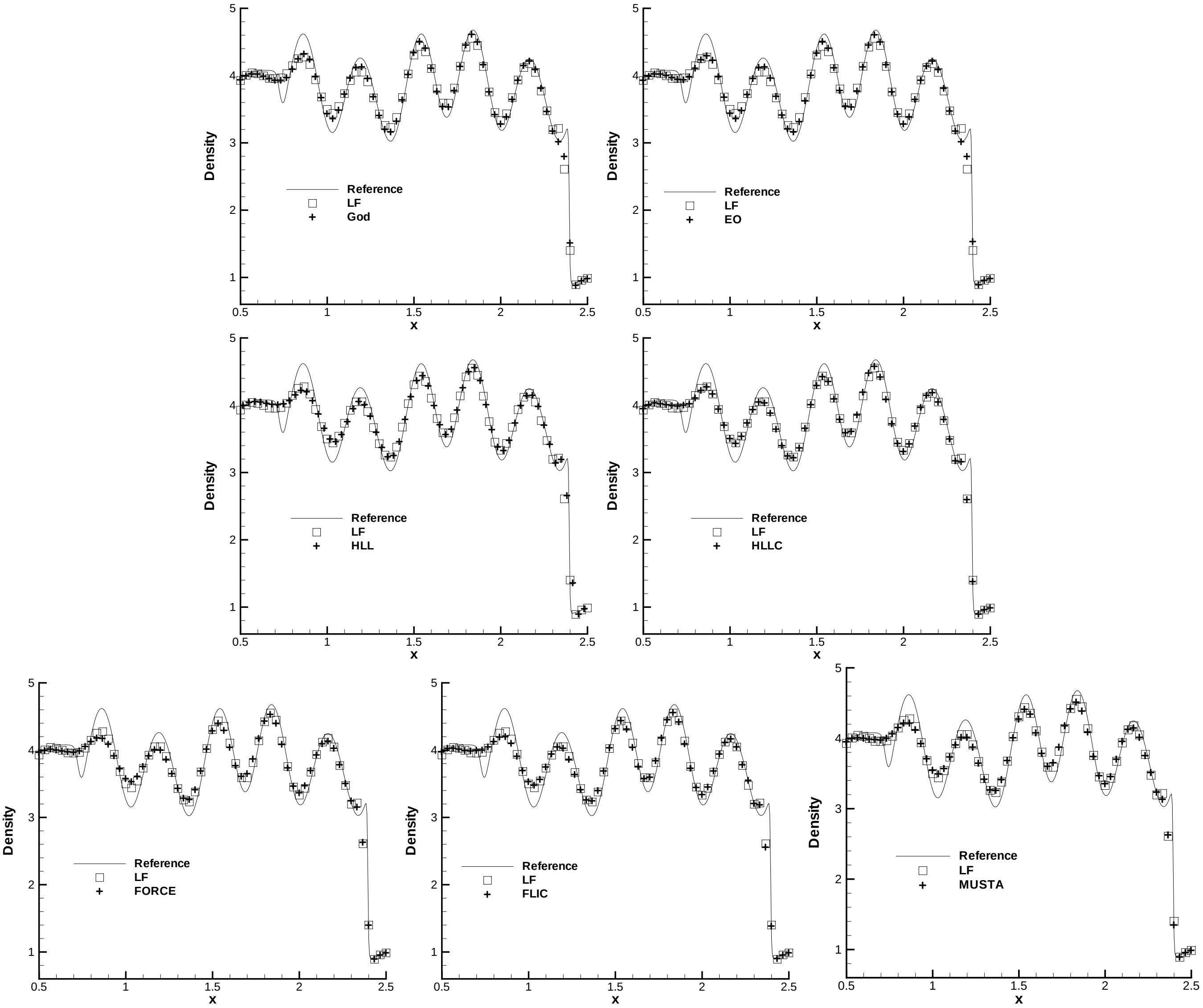

Figure 3.3: Shu-osher problem. $t = 1.3$. HWENO with different fluxes, 200 cells. Density. Solid lines: the exact solution; hollow squares: the results computed by the HWENO-LF scheme; plus symbols: results computed by the HWENO-G (top left), HWENO-EO (top right), HWENO-HLL (middle left), HWENO-HLLC (middle right), HWENO-FORCE (bottom left), HWENO-FLIC (bottom middle) and HWENO-MUSTA (bottom right) schemes.

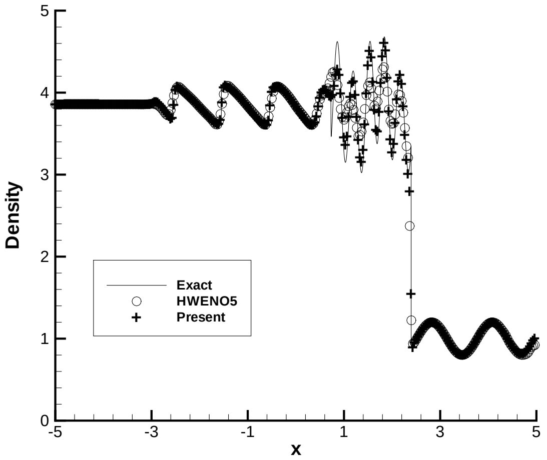

Figure 3.4: The Euler equations Shu-Osher problem. Solid line: the “exact” reference solution; circle: the results computed by classical HWENO5 with LF flux, plus: computed solution of present scheme.

Riemann problem in a computational domain $[0, 1]$ and set the initial conditions as:

$$(\rho, v, p) = \begin{cases} (1, 0, 1000) & \text{if } 0 \leq x < 0.1, \\ (1, 0, 0.01) & \text{if } 0.1 \leq x < 0.9, \\ (1, 0, 100) & \text{if } 0.9 \leq x. \end{cases}$$

A reflecting boundary condition is applied to both ends. We plot the computed densities ρ at $t = 0.038$ with 400 cells, against the reference resolution computed by a fifth-order WENO scheme [8] using 2000 grid points, and against the solution computed by the HWENO-LF scheme on the same mesh. In Figs. 3.5, ”zoomed-in” pictures at the region $[0.53, 0.88]$ show the results which contain the contact discontinuities and shocks in the solution.

Comparing the results from the pictures, the HWENO-LF scheme has the worst resolution among all the schemes. The HWENO-God, HWENO-EO, HWENO-HLL, and HWENO-HLLC have a relatively better resolution than other schemes. Combined with the analysis of the previous examples, HWENO-God and HWENO-EO schemes need more CPU time and have the least resolution, hence we consider the HWENO-HLL and HWENO-HLLC have the better performance in solving the hyperbolic conservation laws based on our schemes.

In Fig. 3.6, we again show the numerical results comparison by our discretization scheme and the classic HWENO-LF method, we can also see that the results by our alternative HWENO-LF scheme have better performance and result than the classic HWENO-LF scheme.

Example 3.5. Double mach reflection problem. In this case, we solve the two dimensional Euler equations in a computational domain of $[0, 4] \times [0, 1]$. This model problem is originated from [27]. A reflection wall lies at the bottom of the domain starting from $x = 1/6$, $y = 0$, making a 60^o angle with the x-axis. The problem is initialized by a right-going Mach 3

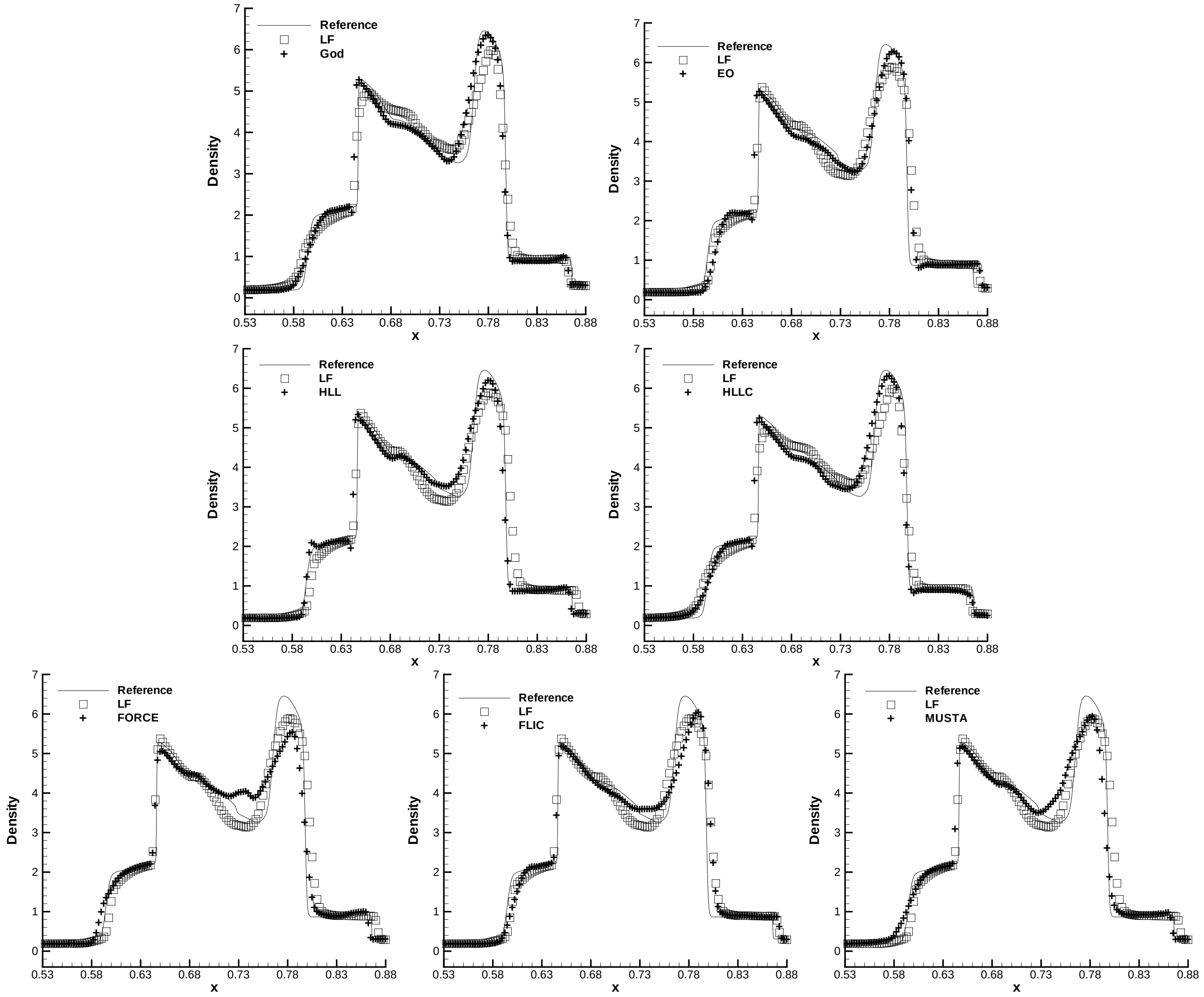

Figure 3.5: Blast wave problem. $t = 1.3$. HWENO with different fluxes, 200 cells. Density. Solid lines: the exact solution; hollow squares: the results computed by the HWENO-LF scheme; plus symbols: results computed by the HWENO-G (top left), HWENO-EO (top right), HWENO-HLL (middle left), HWENO-HLLC (middle right), HWENO-FORCE (bottom left), HWENO-FLIC (bottom middle) and HWENO-MUSTA (bottom right) schemes.

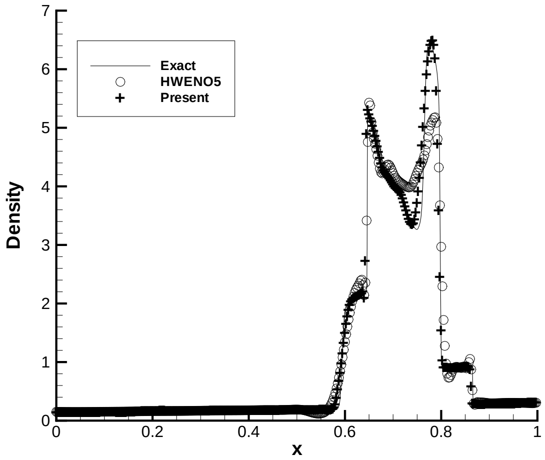

Figure 3.6: The Euler equations Blast-wave problem. Solid line: the "exact" reference solution; circle: the results computed by classical HWENO5 with LF flux, plus: computed solution of present scheme.

flow. The reflection boundary condition is used at the wall, which for the rest of the bottom boundary (the part from $x = 0$ to $x = 1/6$), the exact post-shock condition is imposed. At the top boundary is the exact motion of the mach 10 shock. The results shown are at $t = 0.2$. We present both the pictures of the region $[0, 3] \times [0, 1]$ and the blow-up region around the double mach stems in Figures 3.7. All the figures are showing 30 equally spaced density contours from 1.5 to 22.7. It seems that all schemes perform similarly well for this test case.

According to our numerical experimental results for the one dimensional case, we test only the three relatively better performing schemes, namely the HWENO-LLF, HWENO-HLL, and HWENO-HLLC schemes. In most of the HWENO literature, the Lax-Friedrichs numerical flux is used due to its simplicity. The numerical results of the HWENO-LF scheme also report. It seems that all schemes perform similarly well for this test case.

4 Concluding remarks

In this paper, we study a new form of finite difference HWENO method that the numerical flux framework breaks the limitations of the traditional mathematical form of numerical flux and is suitable for many different forms of numerical flux on the cartesian coordinate. A variety of benchmark examples for one and a two-dimensional hyperbolic system of conservation law are tested. On one hand, extensive numerical results strongly suggest that the present HWENO schemes maintain the excellent properties of the original WENO schemes such as non-oscillatory property near discontinuities, and sharp shock transition. Additionally, the systematically studied and compared a few different fluxes for the HWENO methods make

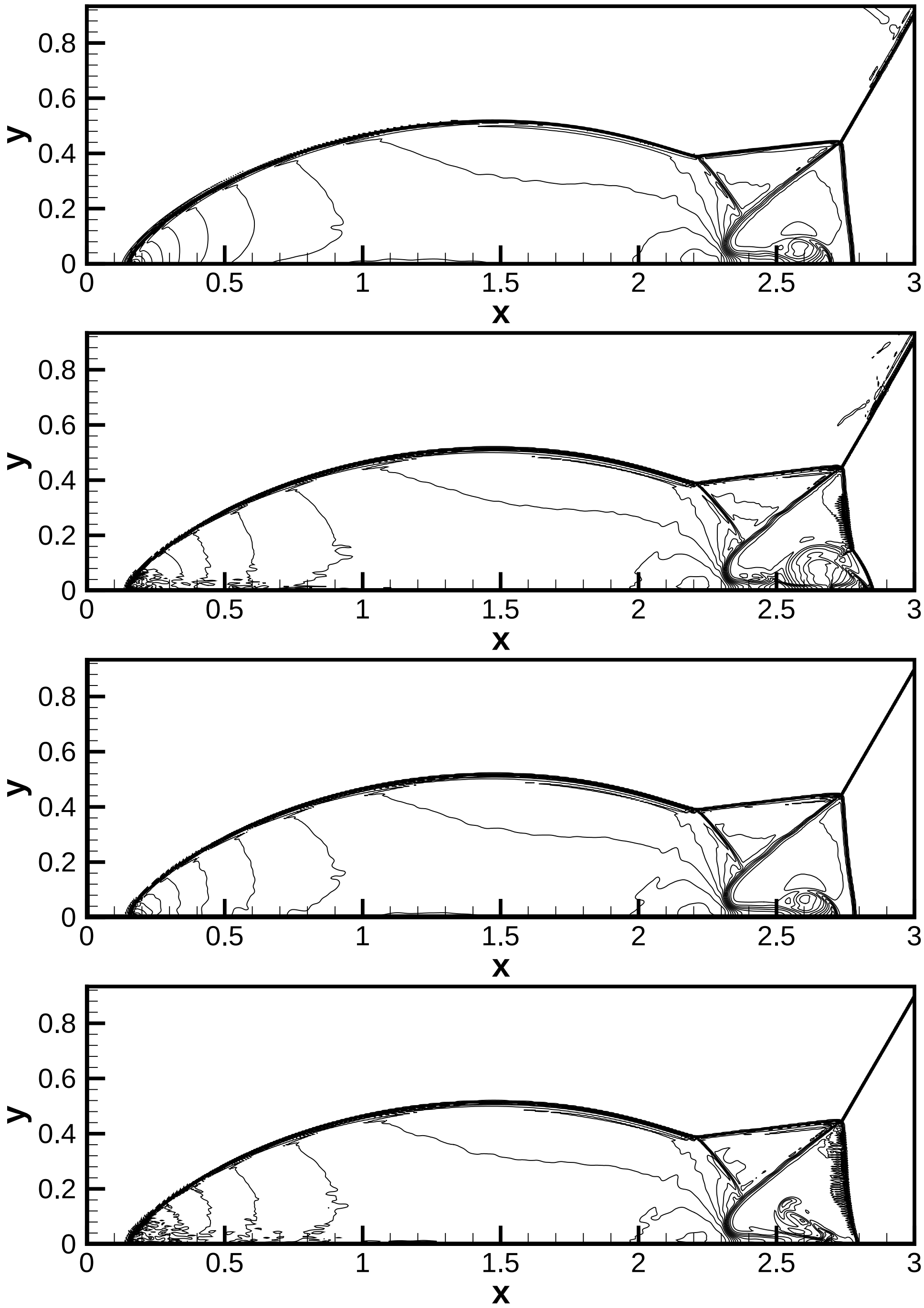

Figure 3.7: Double Mach reflection problem. Density contours for LF,LLF, HLL and HLLC numerical fluxes are shown in panels respectively. 30 density contours are equally spaced between 1.5 to 22.7 on a system size 960 × 240. From top to bottom are HWENO with LF,LLF, HLL and HLLC fluxes,respectively.

us more clearly understand the performance of different fluxes and which flux has better performance for the HWENO methods than the others. Above all, extensive numerical tests indicate that HWENO schemes with the LF flux cost the least CPU time among all, and the numerical errors and resolution of solutions on the discontinuities are also the worst among all. The HWENO-EO and HWENO-Godunov methods seem to cost significantly more CPU time than the HWENO-LF methods. Considering all the factors such as the cost of CPU time, numerical errors and resolution of discontinuities in the solution, the HLL and HLLC fluxes might be good choices as fluxes for the HWENO method. For these good fluxes for HWENO schemes, the numerical test performed for two dimensional systems.

In addition, HWENO schemes have a great advantage over the original WENO schemes in terms of the compactness and the resolutions. Therefore, the HWENO schemes have potential practical applications and the research about the HWENO finite difference schemes for convection-diffusion equations constitutes our ongoing work.

References

[1] M. Ben-Artzi and J. Falcovitz: *A second-order Godunov-type scheme for compressible fluid dynamics*. J. Comput. Phys. 55, 1C32 (1984)

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

[3] B. Cosat and W. S. Don: *High order hybrid central-WENO finite difference scheme for conservation laws*. J. Comput. Appl. Math. 204, 209-218 (2007)

[4] B. Engquist and S. Osher: *One sided difference approximation for nonlinear conservation laws*. Math. Comput. 36, 321-351 (1981)

[5] S. K. Godunov: *Finite difference methods for the computation of discontinuous solutions of the equations of fluid dynamics*. Math. Sbornik 47, 271-306 (1959)

[6] A. Harten, P.D. Lax and B. van Leer: *On upstream differencing and Godunovtype schemes for hyperbolic conservation laws*. SIAM Rev. 25, 35-61 (1983)

[7] D.J. Hill and D.I. Pullin: *Hybrid tuned center-difference-WENO method for large eddy simulations in the presence of strong shocks*. J. Comput. Phys. 194, 435-450 (2004)

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

[9] R.J. LeVeque: *Finite volume methods for hyperbolic problems.* Cambridge University Press (2002)

[10] H. Liu and J. Qiu: *Finite Difference Hermite WENO Schemes for Conservation Laws.* J. Sci. Comput., 63, 548-572 (2015)

[11] S., Osher and F. Solomon: *Upwind difference schemes for hyperbolic conservation laws.* Math. Comput. 38, 339-374 (1982)

[12] S. Pirozzoli: *Conservative hybrid compact-WENO schemes for shock-turbulence interaction.* J. Comput. Phys., 178, 81-117 (2002)

[13] J.Qiu, B.C.Khoo and C.-W. Shu: *A Numerical study for the performance of the Runge-Kutta discontinuous Galerkin method based on different numerical fluxes.* Shock Waves 212, 540-565 (2006)

[14] J. Qiu and C.-W. Shu: *Hermite WENO schemes and their application as limiters for Runge-Kutta Galerkin method: one-dimension case.* J. Comput. Phys. 193, 115-135 (2004)

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

[16] C.-W. Shu: *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws.* NASA/CR-97-206253, ICASE Report NO.97-65 (1997)

[17] C.-W.Shu: *High order weighted essentially nonoscillatory schemes for convection dominated problems.* SIAM Rev. 51, 82-126 (2009)

[18] C.-W. Shu and S. Osher: *Efficient implenmentiation of essentially non-oscillatory shock-capturing schemes.* J. Comput. Phys. 77, 439-471 (1988)

[19] C.-W. Shu and S. Osher: *Efficient implenmentiation of essentially non-oscillatory shock-capturing schemes, II.* J. Comput. Phys. 83, 32-78 (1989)

[20] V.A. Titarev and E.F. Toro: *ADER: arbitrary high-order Godunov approach.* J. Sci. Comput. 17, 609-618 (2002)

[21] V.A. Titarev and E.F. Toro: *Finite-volume WENO schemes for three-dimensional conservation laws.* J. Comput. Phys. 201, 238-260 (2004)

[22] E.F. Toro: *Riemann solvers and numerical methods for fluid dynamics, a practical introduction.* Springer, Berlin (1997)

[23] E.F. Toro: *Shock capturing methods for free surface shallow flows.* Chichester (2001)

[24] E.F. Toro: *Multi-stage predictor-corrector fluxes for hyperbolic equations.* Preprint NI03037-NPA, Isaac Newton Institute for Mathematical Sciences, University of Cambridge, UK.

[25] E.F. Toro: *Riemann solvers and numerical methods for fluid dynamics.* third edition, Springer, Berlin (2009)

[26] E.F. Toro, M. Spruce, and W. Speares: *Restoration of the contact surface in the Harten-Lax-van Leer Riemann solver.* Shock Waves, 4, 25-34 (1994)

[27] P. Woodward and P. Colella: *The numerical simulation of two-dimensional fluid flow with strong shocks.* J. Comput. Phys. 54, 115-173 (1984)