SIAM J. SCI. COMPUT.
Vol. 35, No. 2, pp. A1137–A1160

AN ALTERNATIVE FORMULATION OF FINITE DIFFERENCE WEIGHTED ENO SCHEMES WITH LAX–WENDROFF TIME DISCRETIZATION FOR CONSERVATION LAWS[*]

YAN JIANG[†], CHI-WANG SHU[‡], AND MENGPING ZHANG[†]

Abstract. We develop an alternative formulation of conservative finite difference weighted essentially nonoscillatory (WENO) schemes to solve conservation laws. In this formulation, the WENO interpolation of the solution and its derivatives are used to directly construct the numerical flux, instead of the usual practice of reconstructing the flux functions. Even though this formulation is more expensive than the standard formulation, it does have several advantages. The first advantage 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. The second advantage, which is fully explored in this paper, is that a narrower effective stencil is used compared with previous high order finite difference WENO schemes based on the reconstruction of flux functions, with a Lax–Wendroff time discretization. We will describe the scheme formulation and present numerical tests for one- and two-dimensional scalar and system conservation laws demonstrating the designed high order accuracy and nonoscillatory performance of the schemes constructed in this paper.

Key words. weighted essentially nonoscillatory scheme, Lax–Wendroff time discretization, high order accuracy

AMS subject classification. 65M06

DOI. 10.1137/120889885

1. Introduction.

In this paper, we are interested in designing an alternative formulation of high order conservative finite difference weighted essentially nonoscillatory (WENO) methods, with the Lax–Wendroff time discretizations, in solving nonlinear hyperbolic conservation laws

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

On a uniform mesh $x_i = i\Delta x$, a semidiscrete conservative finite difference scheme for solving (1.1) has the form

$$\frac{d}{dt}u_i + \frac{1}{\Delta x}\left(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}\right) = 0, \tag{1.2}$$

where u_i is an approximation to the point value $u(x_i, t)$, and the numerical flux

$$\hat{f}_{i+\frac{1}{2}} = \hat{f}(u_{i-r}, \dots, u_{i+s})$$

is designed so that

$$\frac{1}{\Delta x}\left(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}\right) = f(u(x))_x|_{x_i} + O(\Delta x^k) \tag{1.3}$$

[*]Submitted to the journal's Methods and Algorithms for Scientific Computing section September 4, 2012; accepted for publication (in revised form) January 2, 2013; published electronically April 23, 2013.

http://www.siam.org/journals/sisc/35-2/88988.html

[†]School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui 230026, People's Republic of China (jiangy@mail.ustc.edu.cn, mpzhang@ustc.edu.cn). The third author's research was supported by NSFC grants 11071234, 91130016, and 91024025.

[‡]Division of Applied Mathematics, Brown University, Providence, RI 02912 (shu@dam.brown.edu). This author's research was supported by AFOSR grant F49550-12-1-0399 and NSF grant DMS-1112700.

for a kth order scheme. Most of the high order finite difference schemes, for example, the high order finite difference WENO schemes in [8, 2, 18], use the following simple Lemma in [20].

Lemma 1.1 (Shu and Osher [20]). *If a function $h(x)$ satisfies the relationship*

$$f(u(x)) = \frac{1}{\Delta x}\int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} h(\xi)d\xi, \tag{1.4}$$

then

$$\frac{1}{\Delta x}\left(h\left(x+\frac{\Delta x}{2}\right) - h\left(x-\frac{\Delta x}{2}\right)\right) = f(u(x))_x. \tag{1.5}$$

The proof of this lemma is a simple application of the fundamental theorem of calculus.

With this simple lemma, it is clear that we should take the numerical flux as

$$\hat{f}_{i+\frac{1}{2}} = h(x_{i+\frac{1}{2}}) + O(\Delta x^k), \tag{1.6}$$

with a smooth coefficient in the $O(\Delta x^k)$ term, to obtain the high order accuracy (1.3). Since the definition of the function $h(x)$ in (1.4) implies

$$\bar{h}_i = \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} h(\xi)d\xi = f(u_i),$$

which is known once we have the point values u_i, we are facing the standard reconstruction problem of knowing the cell averages $\{\bar{h}_i\}$ of the function $h(x)$ and seeking to reconstruct its point values at the cell boundaries $h(x_{i+\frac{1}{2}})$ to high order accuracy, to be used as the numerical flux $\hat{f}_{i+\frac{1}{2}}$ in (1.6). This reconstruction problem is the backbone of the standard finite volume schemes, for example, the ENO finite volume schemes [7] and WENO finite volume schemes [11]. Therefore, using the simple Lemma 1.1, we can simply apply the standard reconstruction subroutine in a high order finite volume scheme to obtain the numerical flux $\{\hat{f}_{i+\frac{1}{2}}\}$ for a high order conservative finite difference scheme from the flux point values $\{f(u_i)\}$.

Because of the clean concept and easy implementation, this formulation of numerical fluxes in high order conservative finite difference schemes has been used widely, for example, in high order finite difference ENO schemes [20] and WENO schemes [8, 2, 18]. However, there are several disadvantages of this formulation:

1. In order to achieve nonlinear stability, an approximate Riemann solver or a two-point monotone flux is used in finite volume schemes, which incorporates upwinding into the schemes. With this finite difference flux formulation, similar upwinding can be implemented through the smooth flux splitting

   $$f(u) = f^+(u) + f^-(u),$$

   where $\frac{d}{du}f^+(u) \geq 0$ (or in system case, this Jacobian matrix has only real and positive eigenvalues) and $\frac{d}{du}f^-(u) \leq 0$ (or in system case, this Jacobian matrix has only real and negative eigenvalues). In order to guarantee high order accuracy, the two functions $f^+(u)$ and $f^-(u)$ should be as smooth functions of u as $f(u)$. The most common flux splitting used in finite difference schemes is therefore the Lax–Friedrichs flux splitting, which is the most diffusive among two-point monotone fluxes.

2. The Lax–Wendroff time discretization procedure [14] can be implemented only through repeated interpolation or reconstruction, resulting in a rather wide effective stencil. We will elaborate on this point later in the paper. Disadvantages of wide stencil include the complication when dealing with boundary conditions.
3. Since the reconstruction is performed directly on the flux values $\{f(u_i)\}$ (or $\{f^+(u_i)\}$ and $\{f^-(u_i)\}$ with flux splitting), not on the point values of the solution $\{u_i\}$, it is more difficult to maintain free stream solutions exactly in curvilinear meshes for multidimensional flow computation. This is because the fluxes in curvilinear coordinates involve metric derivatives, resulting in nonexact cancellations when nonlinear reconstructions are performed for different fluxes. In [12], multidimensional Euler equations are used to show that finite difference WENO schemes could preserve the free stream condition in the Cartesian coordinate system but not in the generalized coordinate system. In [3], Cai and Ladeinde also mentioned the difficulty of maintaining free stream conditions in generalized curvilinear coordinate systems for high order finite difference WENO schemes.

We would like to explore an alternative formulation for constructing numerical fluxes in high order conservative finite difference schemes, originally designed in [19], which involves interpolations directly on the point values of the solution $\{u_i\}$ rather than on the flux values. This alternative formulation, even though less clean and more computationally expensive, does overcome all three disadvantages listed above. We will explore overcoming the first two disadvantages, especially the second one, in detail in this paper.

We now give a brief survey of WENO schemes and Lax–Wendroff time discretization. WENO schemes are high order schemes for approximating hyperbolic conservation laws and other convection dominated partial differential equations. They can produce sharp, nonoscillatory discontinuity transitions and high order accurate resolutions for the smooth part of the solution. The first WENO scheme was introduced in 1994 by Liu, Osher, and Chan in their pioneering paper [11], in which a third order accurate finite volume WENO scheme in one space dimension was constructed. In [8], a general framework is provided to design arbitrary high order accurate finite difference WENO schemes, which are more efficient for multidimensional calculations. Very high order finite difference WENO schemes are documented in [2].

The Lax–Wendroff time discretization procedure, which is also called the Taylor type time discretization, is based on the idea of the classical Lax–Wendroff scheme [9], relying on writing the solution at the next time step by Taylor expansion in time and repeatedly using the partial differential equation (PDE) to rewrite the time derivatives in this Taylor expansion as spatial derivatives. These spatial derivatives can then be discretized by standard approximation procedures, for example, the WENO approximation. We denote u_i^n as the approximation of the point values $u(x_i, t^n)$ and $u^{(r)}$ as the rth order time derivative of u, namely, $u^{(r)} = \frac{\partial^r u}{\partial t^r}$. We also use u', u'', and u''' to denote the first three time derivatives of u. By Taylor expansion, we obtain

$$u(x, t + \Delta t) = u(x, t) + \Delta t u' + \frac{\Delta t^2}{2!} u'' + \frac{\Delta t^3}{3!} u''' + \frac{\Delta t^4}{4!} u^{(4)} + \frac{\Delta t^5}{5!} u^{(5)} + \cdots . \tag{1.7}$$

If we approximate the first k derivatives, we can get kth order accuracy in time. The main idea of the Lax–Wendroff time discretization procedure is to rewrite the time derivatives $u^{(r)}$ as spatial derivatives utilizing the PDE, then discretize these spatial derivatives to sufficient accuracy. Lax–Wendroff time discretization with different types of spatial discretization methods has been introduced in, e.g., [22, 5, 1] for finite volume schemes and [13, 21, 6] for discontinuous Galerkin methods.

However, for a finite difference scheme using the formulation (1.6) and Lemma 1.1 to compute its numerical fluxes, it is more difficult to implement the Lax–Wendroff procedure because we do not have approximations to the spatial derivatives of the solution u at our disposal, only approximations to the spatial derivatives of the flux function $f(u)$. In [14], Qiu and Shu designed a finite difference WENO scheme with Lax–Wendroff time discretization using the formulation (1.6) and Lemma 1.1. To avoid the difficulty mentioned above, they use the approximations to the lower order time derivatives to approximate the higher order ones. As a consequence, the method in [14] involves a rather wide effective stencil. We will provide more details to illustrate this phenomenon. In the procedure to build a fifth order in space and fourth order in time scheme in [14] for solving the one-dimensional scalar equation $u_t + f(u)_x = 0$, the following steps are involved.

Step 1. Using the PDE, we obtain $u' = -f(u)_x$, and the first order time derivative can be obtained by the conservative finite difference WENO approximation (1.3). To obtain a fifth order WENO approximation for $u'|_j^n$, we would need the point values $S = \{u^n_{j-3}, \ldots, u^n_{j+3}\}$.

Step 2. When constructing the second order time derivative $u'' = -(f'(u)u')_x$, we denote $g_j = f'(u_j)u'_j$, where u'_j are the point values of the first order time derivative computed in Step 1. Then we can use a fourth order central difference formula to approximate $u'' = -g_x$ at the point (x_j, t^n):

$$u''_j = -\frac{1}{12\Delta x}(g_{j-2} - 8g_{j-1} + 8g_{j+1} - g_{j+2}).$$

It was observed in [14] that the more costly WENO approximation is not necessary for second and higher order time derivatives, as the standard central difference approximations perform well. Considering $g_j = f'(u_j)u'_j$ and using Step 1 to obtain u'_j, we have the effective stencil for obtaining the point value $u''|_j^n$ to be $S = \{u^n_{j-5}, \ldots, u^n_{j+5}\}$.

Step 3. When constructing the third order time derivative $u''' = -(f'(u)u'' + f''(u)(u')^2)_x$, we let $g_j = (f'(u_j)u''_j + f''(u_j)(u')^2_j)$, where the point values u'_j and u''_j are obtained in Steps 1 and 2, respectively. Then we repeat Step 2 to approximate the third order time derivative using a fourth order central difference approximation. (Third order accuracy is enough here, but central approximations provide only even orders of accuracy.) Considering u'_j and u''_j are obtained in Steps 1 and 2, we obtain the effective stencil for approximating u''' at the point (x_j, t^n) to be $S = \{u^n_{j-7}, \ldots, u^n_{j+7}\}$.

Step 4. The construction of the fourth order time derivative $u^{(4)} = -(f'(u)u''' + 3f''(u)u'u'' + f'''(u)(u')^3)_x$ is obtained in a similar fashion. Let $g_j = f'(u_j)u'''_j + 3f''(u_j)u'_ju''_j + f'''(u_j)(u')^3_j$, where the point values u'_j, u''_j, and u'''_j are obtained in Steps 1, 2, and 3, respectively. Then we use a second order central difference to approximate $u^{(4)}_j = -\frac{1}{2\Delta x}(g_{j+1} - g_{j-1})$. Thus, to get the value $u^{(4)}$ at the point (x_j, t^n), we actually need point values in the effective stencil $S = \{u^n_{j-8}, \ldots, u^n_{j+8}\}$.

In summary, to obtain a fifth order in space and fourth order in time Lax–Wendroff type WENO scheme in [14], the effective stencil is $S = \{I_{j-8}, \ldots, I_{j+8}\}$ for the point x_j, containing 17 grid points. This is a pretty wide stencil. Of course, it is still narrower than Runge–Kutta time discretization. When a four-stage, fourth order Runge–Kutta time discretization is used on a fifth order finite difference WENO scheme, the width of the effective stencil is 25 grid points.

One major motivation to discuss the alternative formulation of high order conservative finite difference schemes in this paper is to make the effective stencil narrower when applying the Lax–Wendroff time discretization. We will review WENO interpolation in section 2, which is the main approximation tool for the alternative formulation of high order conservative finite difference schemes. Details on the construction and implementation of the alternative formulation will be described in section 3 for one- and two-dimensional scalar and system equations. In section 4, extensive numerical examples are provided to demonstrate the accuracy and effectiveness of the methods. Concluding remarks are given in section 5.

2. WENO interpolation.

A major building block of the alternative formulation of the high order conservative finite difference scheme discussed in this paper is the following WENO interpolation procedure. Given the point values $u_i = u(x_i)$ of a piecewise smooth function $u(x)$, we would like to find an approximation of $u(x)$ at the half nodes $x_{i+\frac{1}{2}}$. This WENO interpolation procedure has been developed in, e.g., [16, 4, 18] and will be described briefly below.

The first component of the general WENO produce is a set of lower order approximations to the target value, based on different stencils, which are referred to as the *small stencils*. We would find a unique polynomial of degree at most $k-1$, denoted by $p_r(x)$, which interpolates the function $u(x)$, namely, $p_r(x_j) = u_j$, at the mesh points x_j in the small stencil $S_r = \{x_{i-r}, x_{i-r+1}, \dots, x_{i+s}\}$ for $r = 0, 1, \dots, k-1$ with $r+s+1 = k$. Then, we would use $u^{(r)}_{i+\frac{1}{2}} = p_r(x_{i+\frac{1}{2}})$ as an approximation to the value $u(x_{i+\frac{1}{2}})$, and we have

$$u^{(r)}_{i+\frac{1}{2}} - u(x_{i+\frac{1}{2}}) = O(\Delta x^k) \tag{2.1}$$

if the function $u(x)$ is smooth in the stencil S_r. For example, when $k = 3$, we have $S_0 = \{x_i, x_{i+1}, x_{i+2}\}$, $S_1 = \{x_{i-1}, x_i, x_{i+1}\}$, $S_2 = \{x_{i-2}, x_{i-1}, x_i\}$, and

$$\begin{aligned}
u^{(0)}_{i+\frac{1}{2}} &= \frac{3}{8}u_i + \frac{3}{4}u_{i+1} - \frac{1}{8}u_{i+2},\\
u^{(1)}_{i+\frac{1}{2}} &= -\frac{1}{8}u_{i-1} + \frac{3}{4}u_i + \frac{3}{8}u_{i+1},\\
u^{(2)}_{i+\frac{1}{2}} &= \frac{3}{8}u_{i-2} - \frac{5}{4}u_{i-1} + \frac{15}{8}u_i.
\end{aligned}$$

The second component of the general WENO procedure is to combine the set of lower order approximations to form a higher order approximation to the same target value, based on the *big stencil* $S = \{x_{i-k+1}, \dots, x_{i+k-1}\}$, which is the union of all the small stencils in the first component. We would find a unique polynomial of degree at most $2k-2$, denoted by $p(x)$, interpolating the function $u(x)$ at the points in the big stencil. We use $u_{i+\frac{1}{2}} = p(x_{i+\frac{1}{2}})$ as the approximation to the value $u(x_{i+\frac{1}{2}})$, and we have

$$u_{i+\frac{1}{2}} - u(x_{i+\frac{1}{2}}) = O(\Delta x^{2k-1}) \tag{2.2}$$

provided that the function $u(x)$ is smooth in the big stencil S. When $k = 3$, we have

$$u_{i+\frac{1}{2}} = \frac{3}{128}u_{i-2} - \frac{5}{32}u_{i-1} + \frac{45}{64}u_i + \frac{15}{32}u_{i+1} - \frac{5}{128}u_{i+2}.$$

There are constants d_r such that

$$u_{i+\frac{1}{2}} = \sum_{r=0}^{k-1} d_r u^{(r)}_{i+\frac{1}{2}}, \tag{2.3}$$

where the constants satisfy $\sum_{r=0}^{k-1} d_r = 1$. These constants are usually referred to as the *linear weights* in the WENO procedure. For the case $k = 3$, the linear weights are given as

$$d_0 = \frac{5}{16}, \qquad d_1 = \frac{5}{8}, \qquad d_2 = \frac{1}{16}.$$

The third component of the general WENO procedure is to change the linear weights into nonlinear weights, to ensure both accuracy in smooth cases and non-oscillatory performance when at least one of the small stencils contains a discontinuity of the function $u(x)$. We can achieve this through changing the linear weights d_r into the nonlinear weights ω_r, defined by

$$\omega_r = \frac{\alpha_r}{\sum_{s=0}^{k-1} \alpha_s}, \quad r = 0, \ldots, k-1, \tag{2.4}$$

with

$$\alpha_r = \frac{d_r}{(\epsilon + \beta_r)^2}. \tag{2.5}$$

Here, we take $\epsilon = 10^{-6}$, avoiding letting the denominator be zero, and β_r are the so-called smoothness indicators of the stencil S_r, which measures the smoothness of $u(x)$ in this stencil. Following the original WENO recipe [8], we define

$$\beta_r = \sum_{l=1}^{k-1} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \Delta x^{2l-1} \left(\frac{\partial^l p_r(x)}{\partial^l x} \right)^2 dx. \tag{2.6}$$

When $k = 3$, the formula gives

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

Thus, we get the WENO interpolation of $u(x)$ at the point $x_{i+\frac{1}{2}}$ as

$$u_{i+\frac{1}{2}} = \sum_{r=0}^{k-1} \omega_r u^{(r)}_{i+\frac{1}{2}}. \tag{2.7}$$

We denote it by $u^-_{i+\frac{1}{2}}$ since the big stencil S used to obtain this approximation is biased to the left (there is one more grid point to the left of $x_{i+\frac{1}{2}}$ than to the right). If $u(x)$ is smooth in this big stencil S, we have

$$u^-_{i+\frac{1}{2}} - u(x_{i+\frac{1}{2}}) = O(\Delta x^{2k-1}).$$

If one of the small stencils contains a discontinuity, the corresponding smoothness indicator will be much larger than that of other small stencils; hence the nonlinear weight corresponding to this small stencil will be very small, thus achieving ENO performance. If we choose the big stencil as $S = \{x_{i-k+2}, \ldots, x_{i+k}\}$ which is biased to the right, and the small stencils as $S_r = \{x_{i-r+1}, \ldots, x_{i-r+k}\}$, for $r = 0, \ldots, k-1$, we can obtain in a similar fashion a WENO interpolation of $u(x)$ at the point $x_{i+\frac{1}{2}}$, denoted by $u^+_{i+\frac{1}{2}}$, which has the same high order accuracy in smooth regions and ENO performance near discontinuities as $u^-_{i+\frac{1}{2}}$. In fact, the process to obtain $u^+_{i+\frac{1}{2}}$ is mirror-symmetric to that for $u^-_{i+\frac{1}{2}}$, with respect to the target point $x_{i+\frac{1}{2}}$.

3. Construction of the scheme.

3.1. Construction of the numerical flux in one dimension. Assuming that $f(u)$ is a smooth function of u, we would like to find a consistent numerical flux function

$$\hat{f}_{i+\frac{1}{2}} = \hat{f}(u_{i-r}, \ldots, u_{i+s}) \tag{3.1}$$

such that the flux difference approximates the derivative $f(u(x))_x$ to kth order accuracy,

$$\frac{1}{\Delta x}(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}) = f(u(x))_x|_{x_i} + O(\Delta x^k). \tag{3.2}$$

The numerical flux should satisfy the following conditions:

- $\hat{f}$ is a Lipschitz continuous function in all the arguments;
- $\hat{f}$ is consistent with the physical flux f, that is, $\hat{f}(u, \ldots, u) = f(u)$.

It has been shown in [19] that there exist constants $a_2, a_4, \ldots$ such that

$$\hat{f}_{i+\frac{1}{2}} = f_{i+\frac{1}{2}} + \sum_{k=1}^{[(r-1)/2]} a_{2k} \Delta x^{2k} \left(\frac{\partial^{2k}}{\partial x^{2k}} f \right)_{i+\frac{1}{2}} + O(\Delta x^{r+1}), \tag{3.3}$$

which guarantees $k = r$th order accuracy in (3.2). The coefficients a_{2r} in (3.3) can be obtained through Taylor expansion and the accuracy constraint (3.2). To get an approximation with fifth order accuracy ($k = 5$ in (3.2)), we can use the first three terms given as

$$\hat{f}_{i+\frac{1}{2}} = f_{i+\frac{1}{2}} - \frac{1}{24} \Delta x^2 f_{xx}|_{i+\frac{1}{2}} + \frac{7}{5760} \Delta x^4 f_{xxxx}|_{i+\frac{1}{2}}. \tag{3.4}$$

The first term of the numerical flux in (3.4) is approximated by

$$f_{i+\frac{1}{2}} = h\left(u^-_{i+\frac{1}{2}}, u^+_{i+\frac{1}{2}}\right) \tag{3.5}$$

with the values $u^\pm_{i+\frac{1}{2}}$ obtained by the WENO interpolation discussed in the previous section. The two-argument function h is a monotone flux. It satisfies the following:

- $h(a, b)$ is a Lipschitz continuous function in both arguments.
- $h(a, b)$ is a nondecreasing function in a and a nonincreasing function in b. Symbolically $h(\uparrow, \downarrow)$.
- $h(a, b)$ is consistent with the physical flux f, that is, $h(a, a) = f(a)$.

Examples of the monotone fluxes are the following:

1. the Godunov flux,

$$h(a,b)=\begin{cases}\min_{a\le x\le b} f(u) & \text{if } \ a\le b,\\ \max_{b\le x\le a} f(u) & \text{if } \ a>b;\end{cases} \tag{3.6}$$

2. the Engquist–Osher flux,

$$h(a,b)=\int_0^a \max(f'(u),0)du+\int_0^b \min(f'(u),0)du+f(0); \tag{3.7}$$

3. the Lax–Friedrichs flux,

$$h(a,b)=\frac{1}{2}[f(a)+f(b)-\alpha(b-a)], \tag{3.8}$$

   where $\alpha=\max_u|f'(u)|$ is a constant. The maximum is taken over the relevant range of u.

More discussions on monotone fluxes can be found in, e.g., [10].

The remaining terms of the numerical flux in (3.4) 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, following the practice in [14], 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 WENO procedure. The difference with [14] is that we use chain rules and approximate the values of u and its spatial derivatives, not the derivatives of $f(u)$ directly. For example, when approximating $f_{xx}=f''(u)(u_x)^2+f'(u)u_{xx}$, the point value u and its spatial derivatives u_x and u_{xx} can be obtained by simple central approximation or one-point upwind-biased approximation without the WENO procedure.

3.2. One-dimensional scalar case. Consider the one-dimensional scalar conservation law

$$\begin{cases}u_t+f(u)_x=0,\\ u(x,0)=u_0(x).\end{cases} \tag{3.9}$$

We describe the Lax–Wendroff time discretization to get kth order accuracy in time. In this paper, we will implement the procedure up to fourth order in time.

1. The approximation of the first time derivative $u'=-f(u)_x$ follows the standard WENO procedure. We construct the numerical flux with $(2r-1)$th order accuracy as described in the previous section. In particular, the points $\{x_{i-r+1},\dots,x_{i+r}\}$ are required to obtain the WENO interpolation $u^{\pm}_{i+\frac{1}{2}}$. For the spatial derivatives $\frac{\partial^k f}{\partial x^k}$ at $x_{i+\frac{1}{2}}$, we use $(2r-1-k)$th order simple finite difference approximations based on the same points $\{x_{i-r+1},\dots,x_{i+r}\}$. In our numerical experiments, we aim for fifth order spatial accuracy and take $r=3$. The numerical flux corresponding to u' is then given as

$$\begin{aligned}\hat{f}^{(1)}_{i+\frac{1}{2}}&=h\left(u^-_{i+\frac{1}{2}},u^+_{i+\frac{1}{2}}\right)-\frac{1}{24}\Delta x^2(f''(u)(u_x)^2+f'(u)u_{xx})|_{x_{i+\frac{1}{2}}}\\&\quad+\frac{7}{5760}\Delta x^4(f''''(u)(u_x)^4+6f'''(u)(u_x)^2u_{xx}\\&\quad+4f''(u)u_xu_{xxx}+3f''(u)(u_{xx})^2+f'(u)u_{xxxx})|_{x_{i+\frac{1}{2}}}\end{aligned}$$

such that

$$\frac{\hat{f}^{(1)}_{i+\frac{1}{2}} - \hat{f}^{(1)}_{i-\frac{1}{2}}}{\Delta x} = -u'|_{x_i} + O(\Delta x^4),$$

where $u^{\pm}_{i+\frac{1}{2}}$ in the $h(u^{-}_{i+\frac{1}{2}}, u^{+}_{i+\frac{1}{2}})$ term are obtained through the fifth order WENO interpolation and the values of u, u_x, u_{xx}, u_{xxx}, and u_{xxxx} at $x_{i+\frac{1}{2}}$ in the remaining terms are obtained from the standard polynomial interpolation based on the same points $\{x_{i-2}, \dots, x_{i+3}\}$.

2. The next step is to obtain the approximation of the second time derivative $u'' = -(f'(u)u')_x = ((f'(u))^2 u_x)_x$. We let $g = -(f'(u))^2 u_x$ and repeat the process in step 1. The two major differences from above step 1 are that (i) the approximation can be one order lower in accuracy since there is an extra Δt in the coefficient, and (ii) there is no need to involve WENO in the approximation, also because of the extra Δt in the coefficient. Therefore, we would still only need the interpolation polynomial based on the points $\{x_{i-2}, \dots, x_{i+3}\}$ for all the values of u, u_x, u_{xx}, u_{xxx}, and u_{xxxx} at $x_{i+\frac{1}{2}}$ in the expression

$$\begin{aligned}\hat{f}^{(2)}_{i+\frac{1}{2}} &= g|_{x_{i+\frac{1}{2}}} - \frac{\Delta x^2}{24} g_{xx}|_{x_{i+\frac{1}{2}}} \\ &= -(f'(u))^2 u_x|_{x_{i+\frac{1}{2}}} + \frac{\Delta x^2}{24}(2(f''(u))^2(u_x)^3 + 2f'''(u)f'(u)(u_x)^3 \\ &\quad + 6f''(u)f'(u)u_x u_{xx} + (f'(u))^2 u_{xxx})|_{x_{i+\frac{1}{2}}}\end{aligned}$$

such that

$$\frac{\hat{f}^{(2)}_{i+\frac{1}{2}} - \hat{f}^{(2)}_{i-\frac{1}{2}}}{\Delta x} = -u''|_{x_i} + O(\Delta x^4).$$

3. The approximation of the third time derivative $u''' = -(3f''(u)(f'(u))^2 u_x^2 + (f'(u))^3 u_{xx})_x$ is obtained in a similar fashion. Letting $g = (3f''(u)(f'(u))^2 u_x^2 + (f'(u))^3 u_{xx})$, we can obtain the numerical flux as

$$\begin{aligned}\hat{f}^{(3)}_{i+\frac{1}{2}} = g|_{x_{i+\frac{1}{2}}} - \frac{\Delta x^2}{24} g_{xx}|_{x_{i+\frac{1}{2}}} &= (3f''(u)(f'(u))^2(u_x)^2 + (f'(u))^3 u_{xx})|_{x_{i+\frac{1}{2}}} \\ &- \frac{\Delta x^2}{24}(3f''''(u)(f'(u))^2(u_x)^4 + 18f'''(u)f''(u)f'(u)(u_x)^4 \\ &+ 18f'''(u)(f'(u))^2(u_x)^2 u_{xx} + 6(f''(u))^3(u_x)^4 \\ &+ 36(f''(u))^2 f'(u)(u_x)^2 u_{xx} + 12f''(u)(f'(u))^2 u_x u_{xxx} \\ &+ 9f''(u)(f'(u))^2(u_{xx})^2 + (f'(u))^3 u_{xxxx})|_{x_{i+\frac{1}{2}}},\end{aligned}$$

where the values of u, u_x, u_{xx}, u_{xxx}, and u_{xxxx} at $x_{i+\frac{1}{2}}$ can all be obtained from the interpolation polynomial based on the same points $\{x_{i-2}, \dots, x_{i+3}\}$. We will then have

$$\frac{\hat{f}^{(3)}_{i+\frac{1}{2}} - \hat{f}^{(3)}_{i-\frac{1}{2}}}{\Delta x} = -u'''|_{x_i} + O(\Delta x^3).$$

4. The final step is the approximation of the fourth time derivative $u^{(4)} = (4f'''(u)(f'(u))^3(u_x)^3 + 12(f''(u))^2(f'(u))^2(u_x)^3 + 12f''(u)(f'(u))^3 u_x u_{xxx} +$

$(f'(u))^4u_{xxx})_x$. Letting $g = -(4f'''(u)(f'(u))^3(u_x)^3 + 12(f''(u))^2(f'(u))^2(u_x)^3 + 12f''(u)(f'(u))^3u_xu_{xxx} + (f'(u))^4u_{xxx})$, we can obtain the numerical flux as

$$\hat{f}^{(4)}_{i+\frac{1}{2}} = g|_{x_{i+\frac{1}{2}}} = -(4f'''(u)(f'(u))^3(u_x)^3 + 12(f''(u))^2(f'(u))^2(u_x)^3 \\ + 12f''(u)(f'(u))^3u_xu_{xxx} + (f'(u))^4u_{xxx})|_{x_{i+\frac{1}{2}}},$$

where the values of u, u_x, u_{xx}, u_{xxx}, and u_{xxxx} at $x_{i+\frac{1}{2}}$ can all be obtained from the interpolation polynomial based on the points $\{x_{i-2}, \ldots, x_{i+3}\}$. We will then have

$$\frac{\hat{f}^{(4)}_{i+\frac{1}{2}} - \hat{f}^{(4)}_{i-\frac{1}{2}}}{\Delta x} = -u^{(4)}|_{x_i} + O(\Delta x^2).$$

5. We can form the numerical flux as the sum of all the fluxes above,

$$\hat{f}_{i+\frac{1}{2}} = \hat{f}^{(1)}_{i+\frac{1}{2}} + \frac{\Delta t}{2}\hat{f}^{(2)}_{i+\frac{1}{2}} + \frac{\Delta t^2}{6}\hat{f}^{(3)}_{i+\frac{1}{2}} + \frac{\Delta t^3}{24}\hat{f}^{(4)}_{i+\frac{1}{2}},$$

and obtain the final scheme

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\left(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}\right),$$

which is fifth order accurate in space and fourth order accurate in time.

Of course, we can also increase the time accuracy to fifth order; however, heavier algebra will be involved. In practice, the time accuracy can usually be lower than spatial accuracy to achieve the desired resolution on modest meshes. This is also verified in our numerical results in section 4.

Clearly, the fifth order in space, fourth order in time Lax–Wendroff finite difference scheme constructed above has an effective stencil of only 7 points, the same as the semidiscrete scheme. This compares quite favorably with the Lax–Wendroff type scheme in [14], which has an effective stencil of 17 points, or a fourth order Runge–Kutta finite difference WENO scheme, which has an effective stencil of 25 points.

3.3. One-dimensional syste ms.

For a one-dimensional system of conservation laws (3.9),

$$u(x,t) = (u^1(x,t), \ldots, u^m(x,t))^T \tag{3.10}$$

is a vector and

$$f(u) = (f^1(u^1, \ldots, u^m), \ldots, f^m(u^1, \ldots, u^m))^T \tag{3.11}$$

is a vector function of u. Suppose the system is hyperbolic, i.e., the Jacobian $f'(u)$ has m real eigenvalues

$$\lambda_1(u) \le \cdots \le \lambda_m(u) \tag{3.12}$$

and a complete set of independent eigenvectors

$$r_1(u), \ldots, r_m(u). \tag{3.13}$$

We denote the matrix whose columns are eigenvectors by

$$R(u) = (r_1(u), \ldots, r_m(u)). \tag{3.14}$$

Then

$$R^{-1}(u)f'(u)R(u) = \Lambda(u), \tag{3.15}$$

where $\Lambda(u)$ is the diagonal matrix with $\lambda_1(u), \ldots, \lambda_m(u)$ on the diagonal.

Similar to the scale case, we replace the time derivatives by spatial derivatives using the PDE. We still use the numerical flux for each component

$$\hat{g}_{i+\frac{1}{2}} = g_{i+\frac{1}{2}} + \sum_{k=1}^{[(r-1)/2]} a_{2k}\Delta x^{2k} \frac{\partial^{2k} g}{\partial x^{2k}}|_{i+\frac{1}{2}} + O(\Delta x^{r+1}) \tag{3.16}$$

when approaching g_x.

When constructing the first part of numerical flux for u_t, i.e., $h(u^-_{i+\frac{1}{2}}, u^+_{i+\frac{1}{2}})$ in $\hat{f}^{(1)}_{i+\frac{1}{2}}$, we could use the WENO interpolation procedure on each component of u as in the scalar case. However, for strong shocks this procedure may lead to oscillatory results. Interpolation in the local characteristic fields, even though more costly, may effectively eliminate such spurious oscillations. The procedure of interpolation in the local characteristic fields can be summarized as follows. At each point $x_{i+\frac{1}{2}}$ where the computation of the numerical flux is needed, we perform the following steps:

1. Compute an average state $u_{i+\frac{1}{2}}$, using either the simple arithmetic mean

$$u_{i+\frac{1}{2}} = \frac{1}{2}(u_i + u_{i+1})$$

   or a Roe average [15] satisfying

$$f(u_{i+1}) - f(u_i) = f'(u_{i+\frac{1}{2}})(u_{i+1} - u_i)$$

   if it is available.
2. Compute the right eigenvectors and the left eigenvectors of the Jacobian $f'(u_{i+\frac{1}{2}})$ and denote them by

$$R = R(u_{i+\frac{1}{2}}), \qquad R^{-1} = R^{-1}(u_{i+\frac{1}{2}}).$$

3. Transform all the point values which are in the potential stencil of the WENO interpolation for the obtaining $u^{\pm}_{i+\frac{1}{2}}$ to the local characteristic fields. For our fifth order scheme, we need to compute

$$v_j = R^{-1}u_j, \qquad j = i-2, \ldots, i+3. \tag{3.17}$$

4. Perform the scalar WENO interpolation procedure for each component of the characteristic variable v to obtain the corresponding component of the interpolation $v^{\pm}_{i+\frac{1}{2}}$.
5. Transform back into physical space

$$u^{\pm}_{i+\frac{1}{2}} = Rv^{\pm}_{i+\frac{1}{2}}.$$

For the two-argument flux $h(u^-, u^+)$, instead of the monotone fluxes for the scalar case, we can choose from a variety of numerical fluxes based on exact or approximate Riemann solvers [23]. For example, we can use the Godunov flux, the Lax–Friedrichs flux, or the HLLC flux. This is one of the advantages of the alternative formulation discussed in this paper.

For all remaining terms in the Lax–Wendroff process, we can use simple interpolation polynomials based on the same stencil $\{x_{i-2}, \ldots, x_{i+3}\}$ componentwise to obtain approximations to the values of u, u_x, u_{xx}, u_{xxx}, and u_{xxxx} at $x_{i+\frac{1}{2}}$. No characteristic decomposition or WENO is needed. We remark that in the algebraic manipulations of the Lax–Wendroff process, the derivatives $f'(u)$, $f''(u)$, $f'''(u)$, etc., will become matrices and higher order tensors, increasing the algebraic complexity of the final formulas.

3.4. Two-dimensional cases. Consider the two-dimensional conservation laws:

$$\begin{cases} u_t + f(u)_x + g(u)_y = 0, \\ u(x,y,0) = u_0(x,y). \end{cases} \tag{3.18}$$

The same Taylor expansion still gives

$$u(x,y,t+\Delta t) = u(x,y,t) + \Delta t u' + \frac{\Delta t^2}{2} u'' + \frac{\Delta t^3}{6} u''' + \frac{\Delta t^4}{24} u^{(4)} + \cdots \tag{3.19}$$

and we would still like to use the PDE to replace time derivatives by spatial derivatives. The first time derivative $u' = -f(u)_x - g(u)_y$ can be approximated in a dimension-by-dimension fashion as in standard finite difference schemes [19]. For example, for the fifth order case, we have

$$\begin{aligned} \hat{f}^{(1)}_{i+\frac{1}{2},j} &= h_1(u^-_{i+\frac{1}{2},j}, u^+_{i+\frac{1}{2},j}) - \frac{\Delta x^2}{24}(f''(u)(u_x)^2 + f'(u)u_{xx})|_{(x_{i+\frac{1}{2}},y_j)} \\ &\quad + \frac{7\Delta x^4}{5760}(f''''(u)(u_x)^4 + 6f'''(u)(u_x)^2 u_{xx} + 3f''(u)(u_{xx})^2 \\ &\quad + 4f''(u)u_x u_{xxx} + f'(u)u_{xxxx})|_{(x_{i+\frac{1}{2}},y_j)} \\ \hat{g}^{(1)}_{i,j+\frac{1}{2}} &= h_2(u^-_{i,j+\frac{1}{2}}, u^+_{i,j+\frac{1}{2}}) - \frac{\Delta y^2}{24}(g''(u)(u_y)^2 + g'(u)u_{yy})|_{(x_i,y_{j+\frac{1}{2}})} \\ &\quad + \frac{7\Delta y^4}{5760}(g''''(u)(u_y)^4 + 6g'''(u)(u_y)^2 u_{yy} + 3g''(u)(u_{yy})^2 \\ &\quad + 4g''(u)u_y u_{yyy} + g'(u)u_{yyyy})|_{(x_i,y_{j+\frac{1}{2}})} \end{aligned}$$

such that

$$\begin{aligned} \frac{\hat{f}^{(1)}_{i+\frac{1}{2},j} - \hat{f}^{(1)}_{i-\frac{1}{2},j}}{\Delta x} &= f(u)_x|_{(x_i,y_j)} + O(\Delta x^5), \\ \frac{\hat{g}^{(1)}_{i,j+\frac{1}{2}} - \hat{g}^{(1)}_{i,j-\frac{1}{2}}}{\Delta y} &= g(u)_y|_{(x_i,y_j)} + O(\Delta y^5), \end{aligned}$$

and the values of $u^\pm_{i+\frac{1}{2},j}$, $u^\pm_{i,j+\frac{1}{2}}$ in the monotone fluxes $h_1(\cdot,\cdot)$ (approximating $f(u)$) and $h_2(\cdot,\cdot)$ (approximating $g(u)$) are obtained by one-dimensional WENO interpolation in the same procedure as in the one-dimensional case. The values of u and all its spatial derivatives in all the remaining terms are approximated by simple one-dimensional polynomial interpolations on the same stencil as the WENO approximation, namely, on $S = \{x_{i-2}, \ldots, x_{i+3}\}$ for fixed y_j or on $\{y_{j-2}, \ldots, y_{j+3}\}$ for fixed x_i. This part is also the same as in the one-dimensional case.

For the second time derivative, $u'' = (f'(u)^2 u_x + f'(u)g'(u)u_y)_x + (g'(u)f'(u)u_x + g'(u)^2 u_y)_y$, we can also approximate them in a dimension-by-dimension fashion. For the fifth order scheme, the numerical fluxes have the following form:

$$\begin{aligned}
\hat{f}^{(2)}_{i+\frac{1}{2},j} &= -(f'(u)^2 u_x + f'(u)g'(u)u_y)|_{(x_{i+\frac{1}{2}},y_j)} + \frac{\Delta x^2}{24}(2f'(u)f'''(u)(u_x)^3 \\
&\quad + 2(f''(u))^2(u_x)^3 + 6f'(u)f''(u)u_x u_{xx} + (f'(u))^2 u_{xxx} \\
&\quad + f'''(u)g'(u)(u_x)^2 u_y + 2f''(u)g''(u)(u_x)^2 u_y + f'(u)g'''(u)(u_x)^2 u_y \\
&\quad + f''(u)g'(u)u_{xx}u_y + f'(u)g''(u)u_{xx}u_y + 2f''(u)g'(u)u_x u_{xy} \\
&\quad + 2f'(u)g''(u)u_x u_{xy} + f'(u)g'(u)u_{xxy})|_{(x_{i+\frac{1}{2}},y_j)} \\
\hat{g}^{(2)}_{i,j+\frac{1}{2}} &= -(g'(u)f'(u)u_x + (g'(u))^2 u_y)|_{(x_i,y_{j+\frac{1}{2}})} + \frac{\Delta y^2}{24}(2g'(u)g'''(u)(u_y)^3 \\
&\quad + 2(g''(u))^2(u_y)^3 + 6g'(u)g''(u)u_y u_{yy} + (g'(u))^2 u_{yyy} + g'''(u)f'(u)u_x(u_y)^2 \\
&\quad + 2g''(u)f''(u)u_x(u_y)^2 + g'(u)f'''(u)u_x(u_y)^2 + g''(u)f'(u)u_x u_{yy} \\
&\quad + g'(u)f''(u)u_x u_{yy} + 2g''(u)f'(u)u_y u_{xy} + 2g'(u)f''(u)u_y u_{xy} \\
&\quad + g'(u)f'(u)u_{xyy})|_{(x_i,y_{j+\frac{1}{2}})},
\end{aligned}$$

which satisfies

$$\begin{aligned}
\frac{\hat{f}^{(2)}_{i+\frac{1}{2},j} - \hat{f}^{(2)}_{i-\frac{1}{2},j}}{\Delta x} &= -(f'(u)^2 u_x + f'(u)g'(u)u_y)_x|_{(x_i,y_j)} + O(\Delta x^4) + O(\Delta y^4), \\
\frac{\hat{g}^{(2)}_{i,j+\frac{1}{2}} - \hat{g}^{(2)}_{i,j-\frac{1}{2}}}{\Delta y} &= -(g'(u)f'(u)u_x + g'(u)^2 u_y)_y|_{(x_i,y_j)} + O(\Delta x^4) + O(\Delta y^4).
\end{aligned}$$

All the spatial derivatives can be obtained by using simple one- or two-dimensional polynomial interpolations on the same stencil as the WENO approximation. Two-dimensional standard interpolation on the stencil $S = \{x_{i-2}, \ldots, x_{i+3}\} \times \{y_{j-2}, \ldots, y_{j+3}\}$ is needed to obtain approximations of the mixed derivatives, which have the form

$$\partial u^{r+s}/\partial(x^r y^s)|_{(x_{i+\frac{1}{2}},y_j)\ or\ (x_i,y_{j+\frac{1}{2}})} = \sum_{l=-2}^{3}\sum_{k=-2}^{3} a_k b_l u_{i+k,j+l},$$

where a_k and b_l are constants depending on r, s, Δx, and Δy. This part is different from the one-dimensional case and involves additional cost. Therefore, the Lax–Wendroff procedure may be less attractive than Runge–Kutta methods for time discretization for two- and higher-dimensional nonlinear problems. It should be used only if having a smaller effective stencil is important for the application problems, for example, for better handling of boundary conditions. The higher order time derivatives can be approximated in a similar way.

Once the fluxes for all the time derivatives have been obtained, we can form the final numerical flux as the sum of all these fluxes,

$$\begin{aligned}
\hat{f}_{i+\frac{1}{2},j} &= \hat{f}^{(1)}_{i+\frac{1}{2},j} + \frac{\Delta t}{2}\hat{f}^{(2)}_{i+\frac{1}{2},j} + \frac{\Delta t^2}{6}\hat{f}^{(3)}_{i+\frac{1}{2},j} + \frac{\Delta t^3}{24}\hat{f}^{(4)}_{i+\frac{1}{2},j}, \\
\hat{g}_{i,j+\frac{1}{2}} &= \hat{g}^{(1)}_{i,j+\frac{1}{2}} + \frac{\Delta t}{2}\hat{g}^{(2)}_{i,j+\frac{1}{2}} + \frac{\Delta t^2}{6}\hat{g}^{(3)}_{i,j+\frac{1}{2}} + \frac{\Delta t^3}{24}\hat{g}^{(4)}_{i,j+\frac{1}{2}},
\end{aligned}$$

and obtain the final scheme,

$$u_{i,j}^{n+1} = u_{i,j}^{n} - \frac{\Delta t}{\Delta x}\left(\hat{f}_{i+\frac{1}{2},j} - \hat{f}_{i-\frac{1}{2},j}\right) - \frac{\Delta t}{\Delta y}\left(\hat{g}_{i,j+\frac{1}{2}} - \hat{g}_{i,j-\frac{1}{2}}\right),$$

which is fifth order accurate in space and fourth order accurate in time.

For the system case, WENO interpolation with local characteristic decomposition is still needed for the computation of $u^{\pm}_{i+\frac{1}{2},j}$ and $u^{\pm}_{i,j+\frac{1}{2}}$ in the two-point fluxes $h_1(\cdot,\cdot)$ and $h_2(\cdot,\cdot)$. This part is the same as in the one-dimensional case and uses the information of the Jacobians $f'(u)$ and $g'(u)$ in the x and y directions, respectively. For all other terms, neither local characteristic decomposition nor WENO interpolation is needed. We can use the interpolation polynomials as described above for the scalar case in each component of the solution u.

More details about the Lax–Wendroff discretization procedure can be seen in many references, e.g., for the compressible Euler equations in [5] and for the multidimensional MHD equations in [21].

4. Numerical results. In this section, we present the results of our numerical experiments for the schemes described in the previous section with fourth order in temporal and fifth order in spatial accuracy. In some examples we also compare with the Lax–Wendroff type finite difference WENO scheme in [14], also having fourth order in temporal and fifth order in spatial accuracy. In all our examples, uniform meshes are used, and the CFL number is taken as 0.3. For one-dimensional problems, the time step is chosen as

$$\Delta t^n = 0.3\Delta x/\alpha,$$

where $\alpha = \max_u |f'(u)|$ for the scalar cases and $\alpha = \max_{j=1,\ldots,m} \max_u |\lambda_j(u)|$, where $\lambda_j(u)$ are the eigenvalues of the Jacobian $f'(u)$, for the systems.

Example 1. We first test the accuracy of the scheme on the linear scalar problem

$$u_t + u_x = 0 \tag{4.1}$$

with the initial condition $u(x,0) = \sin(\pi x)$ and 2-periodic boundary condition. The exact solution is $u(x,t) = \sin(\pi(x-t))$. In Table 4.1, we show the errors at time $t = 2$. We can observe fifth order accuracy until we reach quite refined meshes, despite the formal fourth order temporal accuracy, supporting our claim that spatial errors are usually dominating for modest meshes.

Example 2. We test our scheme on the nonlinear Burgers equation

$$u_t + \left(\frac{u^2}{2}\right)_x = 0 \tag{4.2}$$

TABLE 4.1
Accuracy on $u_t + u_x = 0$ with $u(x,0) = \sin(\pi x)$ at $t = 2$.

N_x	L_1 errors	Order	L_∞ error	Order
10	2.38E-02	–	3.67E-02	–
20	1.12E-03	4.41	1.99E-03	4.20
40	3.45E-05	5.02	6.64E-05	4.91
80	1.07E-06	5.00	2.16E-06	4.94
160	3.35E-08	5.00	6.51E-08	5.05
320	1.05E-09	5.00	1.95E-09	5.06
640	3.25E-11	5.01	5.71E-11	5.10

Table 4.2

Accuracy on the Burgers equation with the Lax–Friedrichs flux and the WENO scheme in [14]. $u(x,0) = 0.5 + \sin(\pi x)$ *at* $t = 0.5/\pi$.

	Lax–Friedrichs flux				WENO scheme in [14]			
N_x	L_1 errors	Order	L_∞ error	Order	L_1 errors	Order	L_∞ error	Order
10	4.57E-03	–	1.42E-02	–	3.65E-02	–	6.85E-02	–
20	4.85E-04	3.24	2.33E-03	2.61	4.44E-03	3.04	1.26E-02	2.45
40	2.59E-05	4.23	2.38E-04	3.29	2.55E-04	4.12	1.06E-03	3.57
80	1.37E-06	4.23	1.14E-05	4.38	9.99E-06	4.67	5.11E-05	4.38
160	5.97E-08	4.52	9.90E-07	3.53	3.76E-07	4.73	1.69E-06	4.92
320	1.80E-09	5.05	3.69E-08	4.74	1.15E-08	5.03	7.62E-08	4.47
640	4.07E-11	5.47	3.97E-10	6.54	2.96E-10	5.28	1.46E-09	5.70

Table 4.3

Accuracy on the Burgers equation with the Godunov flux and the Engquist–Osher flux. $u(x,0) = 0.5 + \sin(\pi x)$ *at* $t = 0.5/\pi$.

	Godunov flux				Engquist–Osher flux			
N_x	L_1 errors	Order	L_∞ error	Order	L_1 errors	Order	L_∞ error	Order
10	4.59E-03	–	1.41E-02	–	4.59E-03	–	1.41E-02	–
20	4.84E-04	3.24	2.32E-03	2.60	4.84E-04	3.24	2.32E-03	2.60
40	2.59E-05	4.23	2.38E-04	3.29	2.59E-05	4.23	2.38E-04	3.29
80	1.37E-06	4.23	1.14E-05	4.38	1.37E-06	4.23	1.14E-05	4.38
160	5.97E-08	4.52	9.90E-07	3.53	5.97E-08	4.52	9.90E-07	3.53
320	1.80E-09	5.05	3.69E-08	4.74	1.80E-09	5.05	3.69E-08	4.74
640	4.07E-11	5.47	3.97E-10	6.54	4.07E-11	5.47	3.97E-10	6.54

with the initial condition $u(x,0) = 0.5 + \sin(\pi x)$ and a 2-periodic boundary condition. All the three monotone fluxes mentioned in section 3.1, namely, the Godunov flux, the Engquist–Osher flux, and the Lax–Friedrichs flux, are tested.

When $t = 0.5/\pi$ the solution is still smooth. The errors for all three monotone fluxes are shown in Tables 4.2 and 4.3, respectively. For comparison, we also show the errors by the Lax–Wendroff type finite difference WENO scheme in [14] with the same fourth order in temporal and fifth order in spatial accuracy in Table 4.2. We observe that all schemes achieve fifth order accuracy, despite the formal fourth order temporal accuracy, once again supporting our claim that spatial errors are usually dominating for modest meshes. For this test case, there is very little difference in the errors corresponding to the three different monotone fluxes. We do observe that the errors of the WENO scheme in [14] are significantly bigger than those for the schemes designed in this paper in Tables 4.2 and 4.3 when compared on the same mesh, indicating that the narrower effective stencil and flexibility in choosing numerical fluxes are helping to improve the magnitude of the errors. However, a more meaningful comparison is the CPU cost versus errors achieved. In Figure 4.1, we provide such a comparison for the current scheme with the scheme in [14] for the Lax–Friedrichs flux. Though our method has smaller errors when compared on the same mesh, which is shown in Table 4.2, it seems the two methods have similar efficiency, because of the complex calculation when computing high order time derivatives. This is also a favorable comparison for our method. For systems and multidimensional cases, heavier algebra has a major influence, resulting from matrix calculus and mixed derivatives, and our method has a lower efficiency compared with the scheme in [14]. Our method, when coupled with the Lax–Wendroff time discretization, is therefore of advantage in allowing general monotone fluxes and maintenance of free-stream, rather than in efficiency.

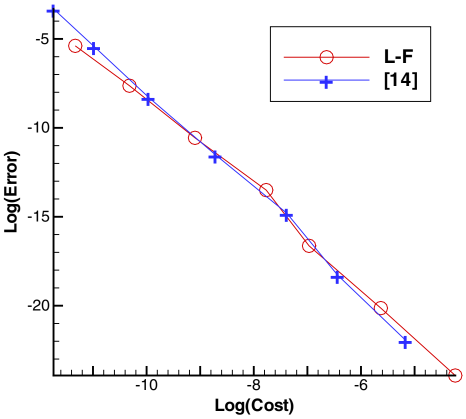


FIG. 4.1. *Comparison of CPU time against L_1 error for one-dimensional Burgers equation between our scheme with the Lax–Friedrichs flux and the scheme in* [14].

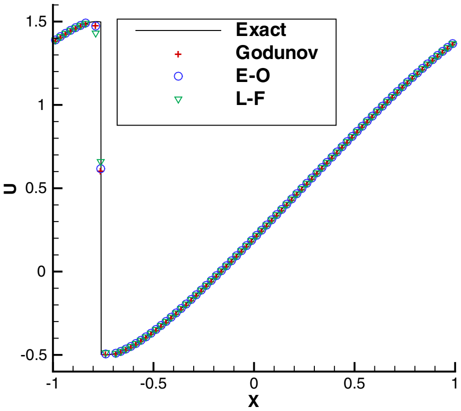


FIG. 4.2. *The Burgers equation with $u(x, 0) = 0.5 + \sin(\pi x)$, $t = 1.5/\pi$, $N = 80$ grid points.*

Next, we plot the solution with 80 grid points at time $t = 1.5/\pi$ in Figure 4.2 when a shock has already appeared in the solution. We can see that for all three monotone fluxes, the schemes simulate the solution with good resolution and without noticeable oscillations near the shock.

Example 3. We now consider the one-dimensional Euler system of compressible gas dynamics:

$$\xi_t + f(\xi)_x = 0, \tag{4.3}$$

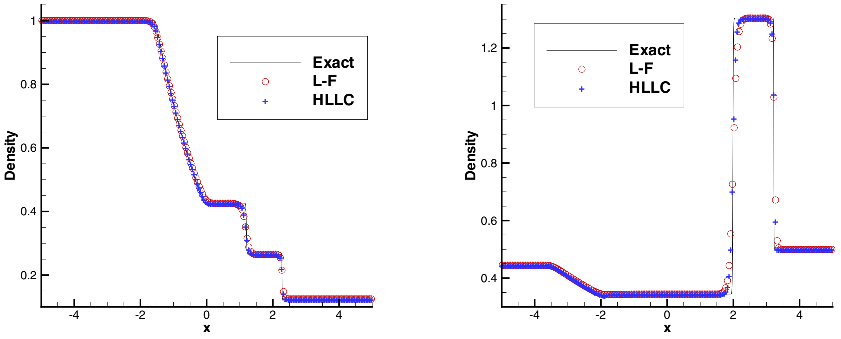
where

$$\xi = (\rho, \rho u, E)^T, \qquad f(\xi) = (\rho u, \rho u^2 + p, u(E + p))^T. \tag{4.4}$$

Here ρ is the density, u is the velocity, 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 $\gamma = 1.4$ for air. The initial condition is set to be $\rho(x, 0) = 1 + 0.2\sin(\pi x)$, $u(x, 0) = 0.7$, $p(x, 0) = 1$, with a 2-periodic boundary condition. The exact solution is $\rho(x, t) = 1 + 0.2\sin(\pi(x - ut))$, $u(x, t) = 0.7$, and $p(x, t) = 1$. The errors and numerical orders of accuracy for the

Table 4.4
Errors and orders of accuracy for the density ρ on the one-dimensional Euler equation at $t = 2$, Lax–Friedrichs flux and HLLC flux.

	Lax–Friedrichs flux				HLLC flux			
N_x	L_1 errors	Order	L_∞ error	Order	L_1 errors	Order	L_∞ error	Order
10	9.20E-03	–	1.30E-02	–	3.46E-03	–	5.58E-03	–
20	4.84E-04	4.25	7.74E-04	4.07	1.55E-04	4.48	2.93E-04	4.25
40	1.53E-05	4.98	2.80E-05	4.79	4.82E-06	5.01	1.00E-05	4.87
80	4.77E-07	5.00	8.89E-07	4.98	1.50E-07	5.01	3.06E-07	5.04
160	1.47E-08	5.02	2.67E-08	5.06	4.63E-09	5.02	8.38E-09	5.19
320	4.42E-10	5.06	7.52E-10	5.15	1.39E-10	5.06	2.37E-10	5.14
640	1.19E-11	5.21	1.99E-11	5.24	3.72E-12	5.22	6.44E-12	5.20



(a) The Sod's problem

(b) The Lax's problem

Fig. 4.3. *The Sod and Lax problems; $t = 1.3$, 200 grid points with the Lax–Friedrichs flux and the HLLC flux.*

density ρ are shown in Table 4.4. We can see that the schemes with both the Lax–Friedrichs flux and the HLLC flux can achieve the designed order of accuracy, while the error magnitude is smaller for the HLLC flux than for the Lax–Friedrichs flux on the same mesh.

Example 4. Next, we solve the same one-dimensional Euler equations with a Riemann initial condition for the Sod problem,

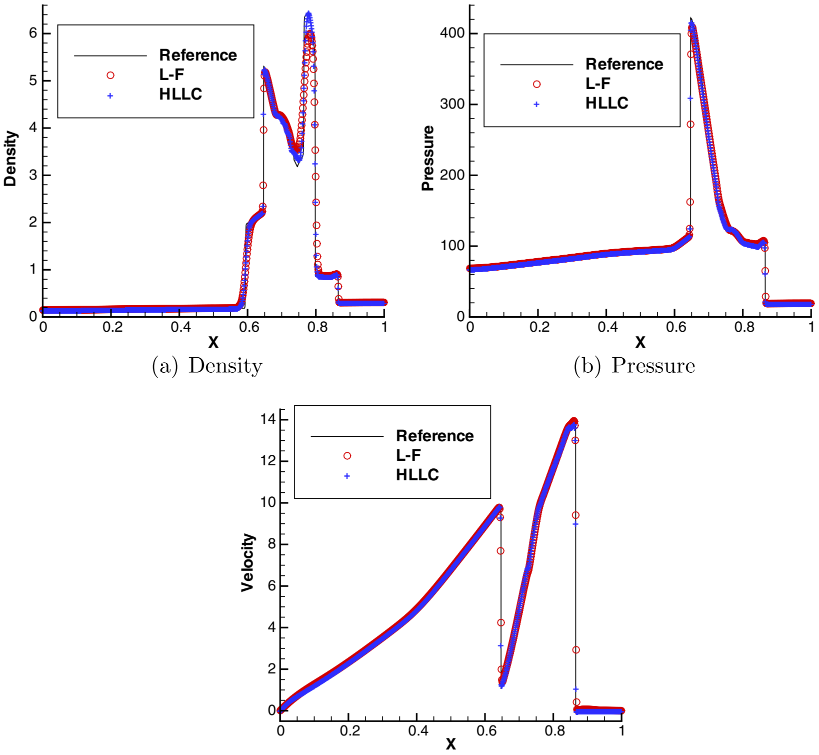
$$(\rho, u, p) = \begin{cases} (1, 0, 1) & \text{if } \ x \le 0, \\ (0.125, 0, 0.1) & \text{if } \ x > 0, \end{cases} \tag{4.5}$$

and the Lax problem,

$$(\rho, u, p) = \begin{cases} (0.445, 0.698, 3.528) & \text{if } \ x \le 0, \\ (0.5, 0, 0.571) & \text{if } \ x > 0. \end{cases} \tag{4.6}$$

We plot the computed density at $t = 1.3$ for both problems with 200 grid points in Figure 4.3. We can see the results of both problems are well resolved and nonoscillatory with both the Lax–Friedrichs flux and the HLLC flux.

Example 5. We plot the numerical solution of the blast wave problem. We are



(a) Density (b) Pressure

(c) Velocity

FIG. 4.4. *Blast wave problem. Lines are from the reference solution, circles are the numerical solution with the Lax–Friedrichs flux, and pluses are the numerical solution with the HLLC flux:* (a) *density,* (b) *pressure,* (c) *velocity.*

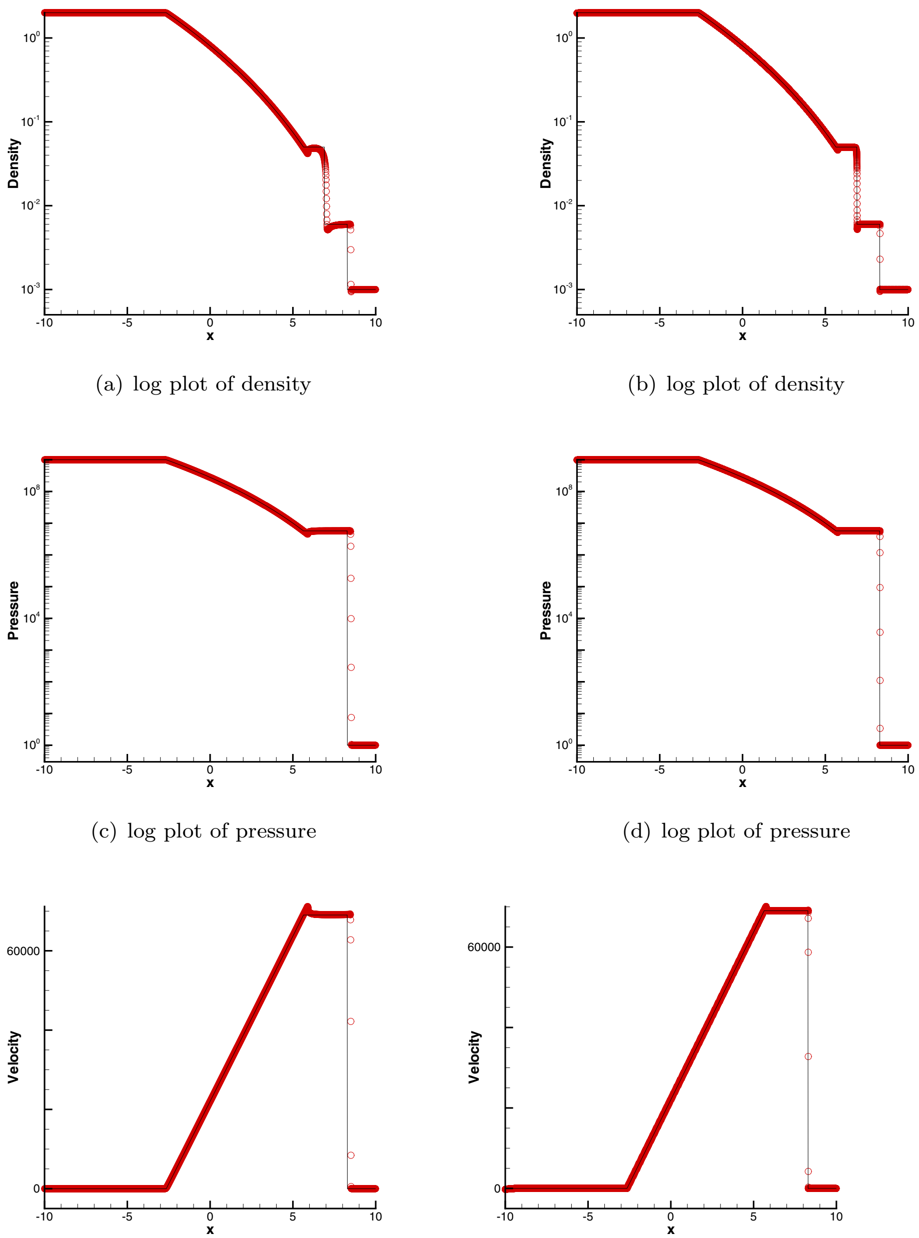
still solving the Euler equations (4.3)–(4.4), with the initial condition

$$
\begin{aligned}
&\rho(x,0) = 1.0, \quad 0 \le x \le 1,\\
&u(x,0) = 0.0, \quad 0 \le x \le 1,\\
&p(x,0) = \begin{cases} 1000 & \text{if} \quad 0 \le x < 0.1,\\ 0.01 & \text{if} \quad 0.1 \le x < 0.9,\\ 100 & \text{if} \quad 0.9 \le x \le 1. \end{cases}
\end{aligned} \tag{4.7}
$$

In Figure 4.4, we compare the numerical solution with a reference solution, which is obtained with the regular WENO scheme in [8] with 16,000 grid points and hence can be considered as the exact solution. We use both the Lax–Friedrichs flux and the HLLC flux for the problem; 800 grid points are used. Our numerical results seem to agree with the reference solution very well.

Example 6. In this example, we solve the LeBlanc shock tube problem, still for the Euler equations (4.3)–(4.4), with the initial condition

$$
\begin{aligned}
&\rho(x,0) = \begin{cases} 2 & \text{if} \quad -10 \le x < 0,\\ 0.001 & \text{if} \quad 0 \le x \le 10, \end{cases}\\
&u(x,0) = 0.0, -10 \le x \le 10,\\
&p(x,0) = \begin{cases} 10^9 & \text{if} \quad -10 \le x < 0,\\ 1 & \text{if} \quad 0 \le x \le 10. \end{cases}
\end{aligned} \tag{4.8}
$$



(a) log plot of density

(b) log plot of density

(c) log plot of pressure

(d) log plot of pressure

(e) velocity

(f) velocity

FIG. 4.5. *LeBlanc problem, $t = 0.0001$. Left: $N_x = 2400$. Right: $N_x = 19200$. The solid line is the exact solution. Symbols are numerical solutions.*

The Lax–Friedrichs flux is used. We plot the numerical solutions with 2400 and 19,200 grid points in Figure 4.5 to compare with the results in [24], in which a third order discontinuous method with 800 cells and 6400 cells is used and each cell contains three degrees of freedom. We can see that our results are well resolved and agree well with the solutions in [24]. This is an extreme test case with very strong discontinuities and is used to test robustness of our scheme.

Example 7. We now consider the following two-dimensional linear equation

$$u_t + au_x + bu_y = 0 \tag{4.9}$$

with the initial condition $u(x, y, 0) = \sin(\pi(x + y))$ and a 2-periodic boundary condition, where a and b are constants. Here, we take $a = 1$, $b = -2$, and the final time $t = 2$. To avoid possible error cancellations due to symmetry, we use different mesh sizes in the x and y directions. Errors are shown in Table 4.5. We can clearly observe that the scheme achieves the designed fifth order accuracy.

Table 4.5

Accuracy on the linear equation $u_t + u_x - 2u_y = 0$ with $u(x, y, 0) = \sin(\pi(x + y))$ at $t = 2$.

$N_x \times N_y$	L_1 errors	Order	L_∞ error	Order
8×12	4.97E-02	–	7.23E-02	–
16×24	5.70E-03	3.12	1.02E-02	2.83
32×48	1.70E-04	5.07	3.42E-04	4.90
64×96	4.72E-06	5.17	1.01E-05	5.08
128×192	1.41E-07	5.07	3.07E-07	5.04
256×384	4.32E-09	5.02	9.16E-09	5.06

Example 8. Next, we solve the two-dimensional nonlinear Burgers equation

$$u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0 \tag{4.10}$$

with the initial condition $u(x, 0) = 0.5 + \sin(\pi(x + y)/2)$ and a 4-periodic boundary condition. When $t = 0.5/\pi$ the solution is still smooth, and we give the errors and order of accuracy in Table 4.6 with the Lax–Friedrichs flux. Clearly, the scheme delivers the designed order of accuracy. For comparison, we also show the errors by the Lax–Wendroff type finite difference WENO scheme in [14] with the same fourth order in temporal and fifth order in spatial accuracy in Table 4.6. We again observe that the errors of the WENO scheme in [14] are significantly bigger than those for the schemes designed in this paper in Table 4.6 when compared on the same mesh, indicating that the narrower effective stencil is helping in improving the magnitude of the errors (at the price of higher computational cost).

We plot the results at $t = 1.5/\pi$ when shocks have already appeared in Figure 4.6 with 160×160 grid points. On the left we show a one-dimensional cut at $x = y$ of the solution, and on the right we show the surface of the computed solution. Clearly, the shocks are well resolved without spurious oscillations.

Example 9. We now consider two-dimensional Euler systems of compressible gas dynamics:

$$\xi_t + f(\xi)_x + g(\xi)_y = 0 \tag{4.11}$$

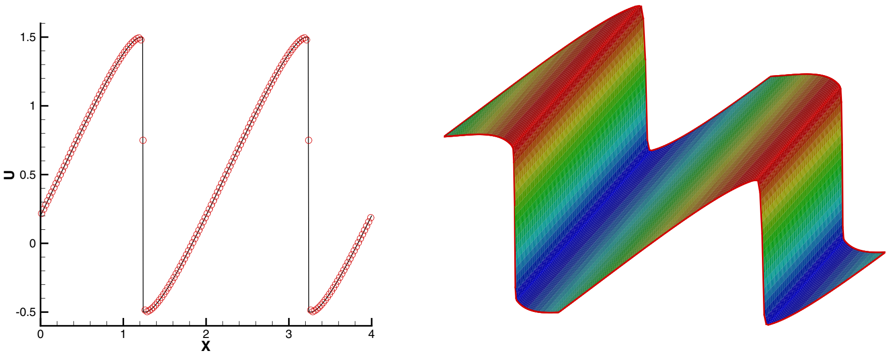
with

$$\begin{aligned} \xi &= (\rho, \rho u, \rho v, E)^T, \\ f(\xi) &= (\rho u, \rho u^2 + p, \rho uv, u(E + p))^T, \\ g(\xi) &= (\rho v, \rho uv, \rho v^2 + p, v(E + p))^T. \end{aligned}$$

TABLE 4.6

Accuracy on the Burgers equation $u_t + (\frac{u^2}{2})_x + (\frac{u^2}{2})_y = 0$ with Lax–Friedrichs flux and the WENO scheme in [14]. $u(x, y, 0) = 0.5 + \sin(\pi(x + y)/2)$ *at* $t = 0.5/\pi$.

	Lax–Friedrichs flux				WENO scheme in [14]			
$N_x \times N_y$	L_1 errors	Order	L_∞ error	Order	L_1 errors	Order	L_∞ error	Order
8×12	7.51E-03	–	2.35E-02	–	2.23E-02	–	6.44E-02	–
16×24	9.86E-04	2.93	6.96E-03	1.75	2.98E-03	2.90	1.45E-02	2.15
32×48	8.23E-05	3.58	7.30E-04	3.25	2.90E-04	3.36	1.65E-03	3.13
64×96	4.26E-06	4.27	4.09E-05	4.16	8.58E-06	5.08	8.08E-05	4.36
128×192	1.84E-07	4.53	1.62E-06	4.65	3.27E-07	4.71	2.57E-06	4.97
256×384	6.24E-09	4.88	7.20E-08	4.49	1.01E-08	5.01	1.33E-07	4.28



(a)

(b)

FIG. 4.6. *The two-dimensional Burgers equation,* $u(x, 0) = 0.5 + \sin(\pi(x + y)/2)$, $t = 1.5/\pi$, 160×160 *grid points. Left: a cut of the solution at* $x = y$, *where the solid line is the exact solution and the circles are the computed solution. Right: the surface of the solution.*

Here, ρ is the density, (u, v) is velocity, E is the total energy, and p is the pressure, related to the total energy E by $E = \frac{p}{\gamma-1} + \frac{1}{2}\rho(u^2 + v^2)$, with $\gamma = 1.4$. The initial condition is set to be $\rho(x, y, 0) = 1 + 0.2\sin(\pi(x+y))$, $u(x, y, 0) = 0.7$, $v(x, y, 0) = 0.3$, $p(x, y, 0) = 1$, with a 2-periodic boundary condition. The exact solution is $\rho(x, y, t) = 1 + 0.2\sin(\pi(x + y - (u + v)t))$, $u = 0.7$, $v = 0.3$, $p = 1$. The errors and order of accuracy for the density are shown in Table 4.7. We can see that the scheme achieves the designed order of accuracy.

Example 10. Next, we use the following problem to illustrate further the high order accuracy of the method. Consider the following idealized problem for the Euler equation in two dimensions. The mean flow is $\rho = 1$, $p = 1$, and $(u, v) = (1, 1)$ (diagonal flow). We add to this mean flow an isentropic vortex (perturbation in (u, v) and the temperature $T = \frac{p}{\rho}$, no perturbation in the entropy $S = \frac{p}{\rho^\gamma}$):

$$\begin{aligned}(\delta u, \delta v) &= \frac{\epsilon}{2\pi} e^{0.5(1-r^2)}(-\bar{y}, \bar{x}),\\ \delta T &= -\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2} e^{1-r^2},\\ \delta S &= 0,\end{aligned}$$

where $(\bar{x}, \bar{y}) = (x-5, y-5)$, $r^2 = \bar{x}^2 + \bar{y}^2$, and the vortex strength $\epsilon = 5$ [17]. Since the mean flow is in the diagonal direction, the vortex movement is not aligned with the

Table 4.7
Accuracy on the two-dimensional Euler equation at $t = 2$, errors of the density ρ.

$N_x \times N_y$	L_1 errors	Order	L_∞ error	Order
8×12	2.29E-02	–	3.28E-02	–
16×24	1.47E-03	3.97	2.22E-03	3.89
32×48	5.15E-05	4.83	9.03E-05	4.62
64×96	1.61E-06	5.00	2.99E-06	4.92
128×192	4.98E-08	5.01	9.20E-08	5.02
256×384	1.51E-09	5.04	2.58E-09	5.15

Table 4.8
Accuracy on the vortex evolution at $t = 0.2$, errors of the density ρ.

$N_x \times N_y$	L_1 errors	Order	L_∞ error	Order
40×40	5.52E-05	–	1.98E-03	–
80×80	2.90E-06	4.25	1.41E-04	3.81
160×160	6.84E-08	5.41	2.57E-06	5.78
320×320	1.96E-09	5.13	5.53E-08	5.54

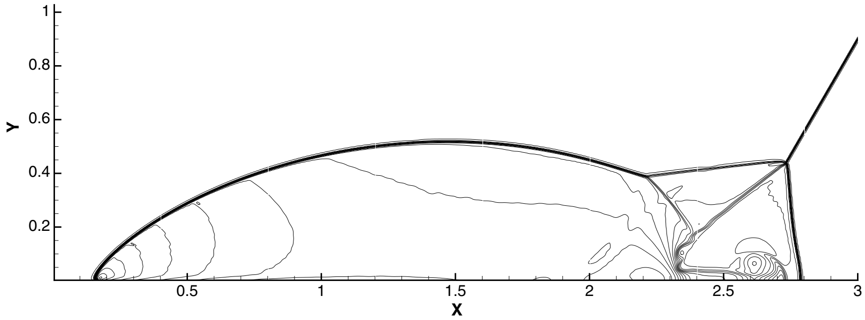


Fig. 4.7. *Double Mach reflection problem;* 30 *equally spaced contours form* 1.5 *for* 22.7 *for the density* ρ.

mesh direction. It is clear that the exact solution is just the passive convection of the vortex with the mean velocity. The computational domain is taken as $[0, 10] \times [0, 10]$, extended periodically in both directions. The simulation is performed until $t = 0.2$. The errors and orders of accuracy are shown in Table 4.8.

Example 11. Finally, we consider the double Mach reflection problem. The

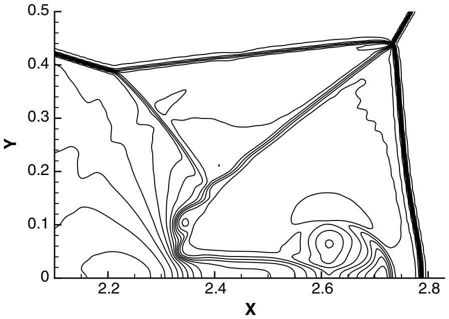


Fig. 4.8. *Zoomed-in figure, double Mach reflection problem;* 30 *equally spaced contours form* 1.5 *for* 22.7 *for the density* ρ.

computational domain for this problem is chosen to be $[0,4]\times[0,1]$. The reflection wall lies at the bottom of the computational domain starting from $x=\frac{1}{6}$. Initially a right-moving Mach 10 shock is positioned at $x=\frac{1}{6}$, $y=0$, and makes a 60° angle with the x-axis. For the bottom boundary, the exact postshock condition is imposed for the part from $x=0$ to $x=\frac{1}{6}$ and a reflective boundary condition is used for the rest. At the top boundary of the computational domain, the flow values are set to describe the exact motion of the Mach 10 shock. The initial preshock condition is

$$(\rho,p,u,v)=(8,\ 116.5,\ 8.25\cos(30^\circ),\ -8.25\sin(30^\circ)) \tag{4.12}$$

and the postshock condition is

$$(\rho,p,u,v)=(1.4,\ 1,\ 0,\ 0). \tag{4.13}$$

We compute the solution up to $t=0.2$. Uniform meshes with grid points 960×239 are used. In Figure 4.7 we show 30 equally spaced density contours from 1.5 to 22.7, and a "zoomed-in" graph is provided in Figure 4.8. We can see that our results agree well with the results in [14].

5. Concluding remarks. We have discussed in depth an alternative formulation for constructing high order conservative finite difference schemes. In this formulation, the high order interpolation procedure is applied to the solution itself rather than to the flux functions. Even though the algebra and algorithm complexity are increased compared with the standard finite difference formulation, this alternative formulation does have several advantages, including the flexibility in using any monotone fluxes in the scalar case and any approximate Riemann solvers in the system case, and a narrow stencil which is the same as that for the semidiscrete scheme when applying the Lax–Wendroff time discretization. We demonstrate these two advantages and discuss the second advantage in depth. We describe the Lax–Wendroff time discretization of this alternative formulation of high order conservative finite difference schemes in detail, using fifth order WENO interpolation for the leading term and fixed stencil polynomial interpolation for the remaining terms. Temporal accuracy is kept at fourth order in the extensive numerical results, including one- and two-dimensional scalar problems and systems. These numerical examples verify the performance of the designed schemes in their designed high order of accuracy and nonoscillatory shock transitions for discontinuous solutions. Another advantage of this alternative formulation, namely, the easier maintenance of free-stream solutions in multidimensional curvilinear coordinates, will be explored in future work.

REFERENCES

[1] D. S. Balsara, T. Rumpf, M. Dumbser, and C. D. Munz, *Efficient, high accuracy ADER-WENO schemes for hydrodynamics and divergence-free magnetohydrodynamics*, J. Comput. Phys., 228 (2009), pp. 2480–2516.

[2] D. Balsara and C.-W. Shu, *Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy*, J. Comput. Phys., 160 (2000), pp. 405–452.

[3] X. Cai and F. Ladeinde, *Performance of WENO Scheme in Generalized Curvilinear Coordinate Systems*, In Proceedings of the 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 2008.

[4] E. Carlini, R. Ferretti, and G. Russo, *A weighted essentially nonoscillatory, large time-step scheme for Hamilton-Jacobi equations*, SIAM J. Sci. Comput., 27 (2005), pp. 1071–1091.

[5] M. Dumbser, M. Käser, V. A. Titarev, and E. F. Toro, *Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems*, J. Comput. Phys., 226 (2007), pp. 204–243.

[6] G. Gassner, F. Löercher, and C. D. Munz, *A discontinuous Galerkin scheme based on a space-time expansion* II. *Viscous flow equations in multi dimensions*, J. Sci. Comput., 34 (2008), pp. 260–286.

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

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

[9] P. D. Lax and B. Wendroff, *Systems of conservation laws*, Commun. Pure Appl. Math., 13 (1960), pp. 217–237.

[10] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhäuser-Verlag, Basel, 1990.

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

[12] T. Nonomura, N. Iizukab, and K. Fujii, *Freestream and vortex preservation properties of high-order WENO and WCNS on curvilinear grids*, Comput. & Fluids, 39 (2010), pp. 197–214.

[13] J. Qiu, M. Dumbser, and C.-W. Shu, *The discontinuous Galerkin method with Lax–Wendroff type time discretizations*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 4528–4543.

[14] J. Qiu and C.-W. Shu, *Finite difference WENO schemes with Lax–Wendroff type time discretizations*, SIAM J. Sci. Comput., 24 (2003), pp. 2185–2198.

[15] P. Roe, *Approximate Riemann solvers, parameter vectors and difference schemes*, J. Comput. Phys., 27 (1978), pp. 1–31.

[16] K. Sebastian and C.-W. Shu, *Multi domain WENO finite difference method with interpolation at sub-domain interfaces*, J. Sci. Comput., 19 (2003), pp. 405–438.

[17] C.-W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, in Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lecture Notes in Math. 1697, Springer, Berlin, 1998, pp. 325–432.

[18] C.-W. Shu, *High order weighted essentially non-oscillatory schemes for convection dominated problems*, SIAM Rev., 51 (2009), pp. 82–126.

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

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

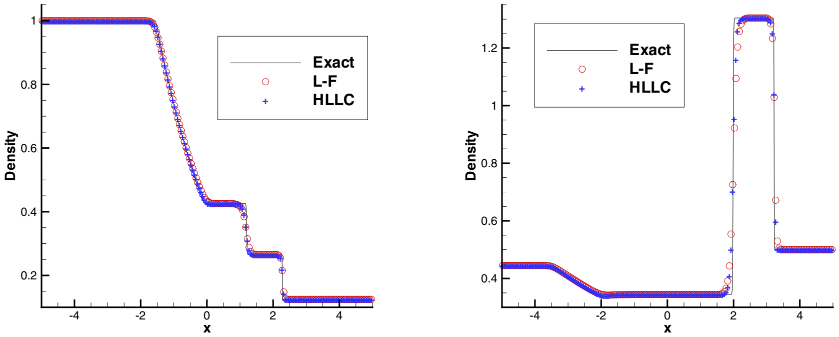
[21] A. Taube, M. Dumbser, D. Balsara, and C.-D. Munz, *Arbitrary high order discontinuous Galerkin schemes for the magnetohydrodynamic equations*, J. Sci. Comput., 30 (2007), pp. 441–464.

[22] V. A. Titarev and E. F. Toro, *ADER schemes for three-dimensional nonlinear hyperbolic systems*, J. Comput. Phys., 204 (2005), pp. 715–736.

[23] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics, a Practical Introduction*, Springer, Berlin, 1997.

[24] X. Zhang and C.-W. Shu, *On positive-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes*, J. Comput. Phys., 229 (2010), pp. 8918–8934.

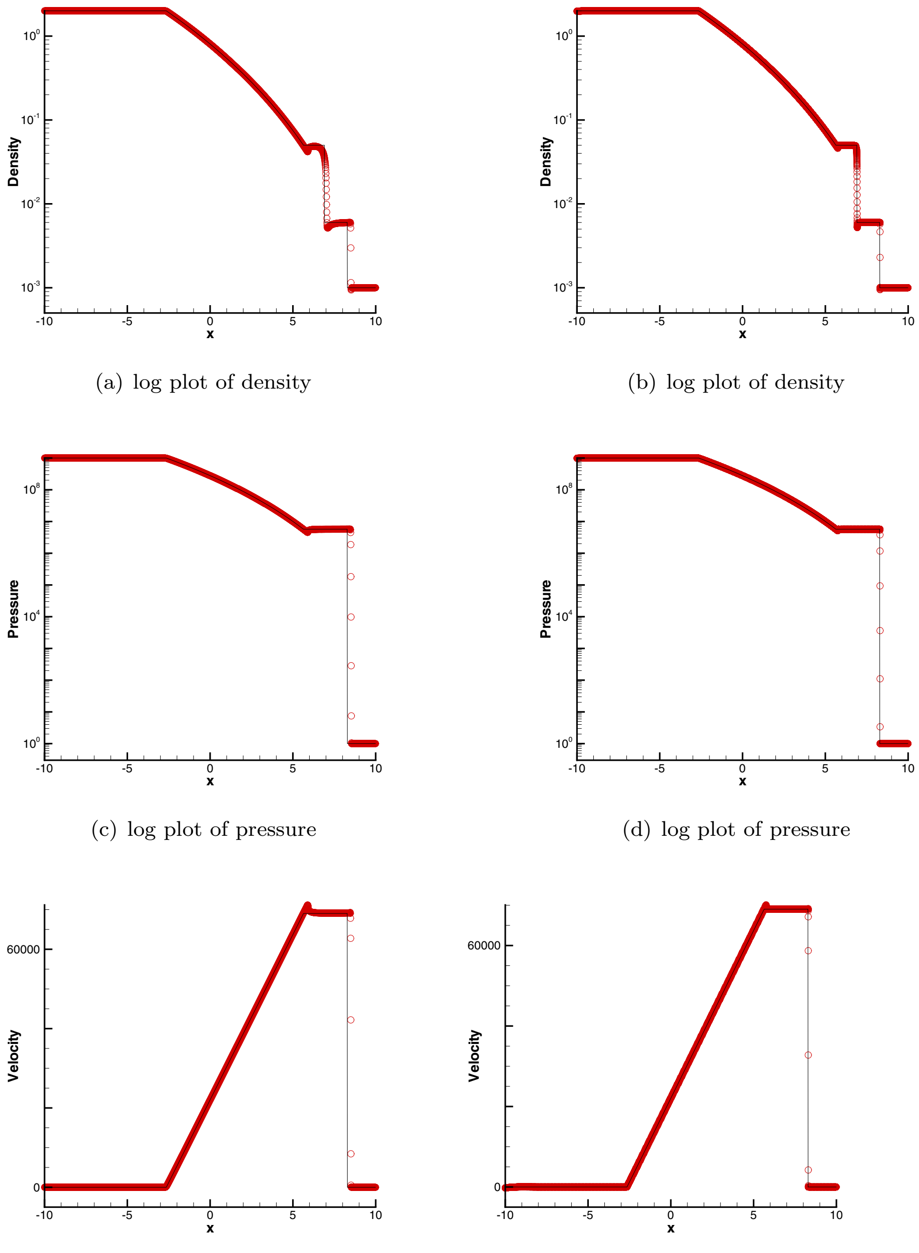
Because of errors in production, Figures 4.3, 4.5, and 4.6 are incorrect. The correct versions are shown below.



(a) The Sod's problem

(b) The Lax's problem

Fig. 4.3. *The Sod and Lax problems;* $t = 1.3$, 200 *grid points with the Lax–Friedrichs flux and the HLLC flux.*



(a) log plot of density

(b) log plot of density

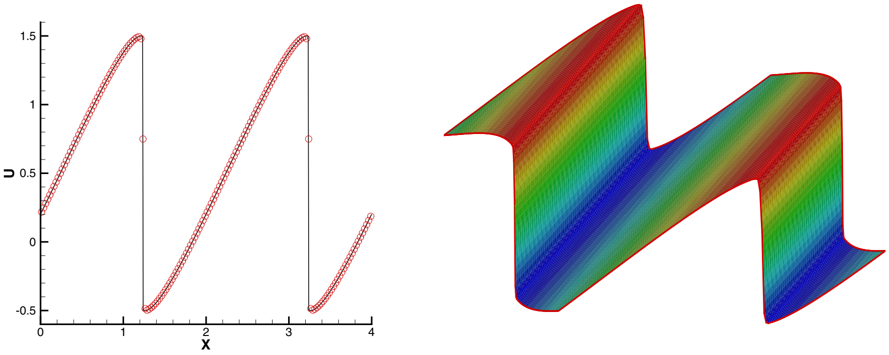
(c) log plot of pressure

(d) log plot of pressure

(e) velocity

(f) velocity

FIG. 4.5. *LeBlanc problem, $t = 0.0001$. Left: $N_x = 2400$. Right: $N_x = 19200$. The solid line is the exact solution. Symbols are numerical solutions.*



(a)

(b)

FIG. 4.6. *The two-dimensional Burgers equation,* $u(x, 0) = 0.5 + \sin(\pi(x + y)/2)$*,* $t = 1.5/\pi$*,* 160×160 *grid points. Left: a cut of the solution at* $x = y$*, where the solid line is the exact solution and the circles are the computed solution. Right: the surface of the solution.*