Journal of Computational Physics 281 (2015) 148–176

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

High-order central Hermite WENO schemes on staggered meshes for hyperbolic conservation laws [☆]

Zhanjing Tao [a], Fengyan Li [b], Jianxian Qiu [a,*]

[a] School of Mathematical Sciences and Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computation, Xiamen University, Xiamen, Fujian 361005, PR China
[b] Department of Mathematical Science, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180, United States

ARTICLE INFO

Article history:
Received 5 August 2014
Received in revised form 11 October 2014
Accepted 13 October 2014
Available online 22 October 2014

Keywords:
Finite volume method
Central schemes
Hermite WENO
Lax–Wendroff
Natural continuous extension (NCE) of Runge–Kutta
High-order accuracy

ABSTRACT

In this paper, we propose a class of high-order schemes for solving one- and two-dimensional hyperbolic conservation laws. The methods are formulated in a central finite volume framework on staggered meshes, and they involve Hermite WENO (HWENO) reconstructions in space, and Lax–Wendroff type discretizations or the natural continuous extension of Runge–Kutta methods in time. Compared with central WENO methods, the spatial reconstruction used here is much more compact; and unlike the original HWENO methods, our proposed schemes require neither flux splitting nor the use of numerical fluxes. In the system case, local characteristic decomposition is applied in the reconstructions of cell averages to enhance the non-oscillatory property of the methods. The high resolution and robustness of the methods in capturing smooth and non-smooth solutions are demonstrated through a collection of one- and two-dimensional scalar and system of examples.

1. Introduction

Hyperbolic conservation laws arise in a wide range of applications in science and engineering, such as aerodynamics, meteorology and weather prediction, astrophysical modeling, multi-phase flow problems, and the study of explosion and blast waves. In general the exact solutions of such equations are not available, and they can also develop discontinuous features, e.g. shocks, compound waves, etc., regardless of the smoothness of the initial and boundary data. It has been an active and important research area to design accurate and robust methods for numerically simulating hyperbolic conservation laws.

In this paper, we design high-order central Hermite WENO (weighted essentially non-oscillatory, C-HWENO) schemes for solving one- and two-dimensional hyperbolic conservation laws

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

[☆] The research was partially supported by NSFC grants 91230110 and 11328104, NSF grants DMS-0847241 and DMS-1318409, and the Fundamental Research Funds for Xiamen University, Grant No. 201412G005.

[*] Corresponding author.
E-mail addresses: tzjnchy555@163.com (Z. Tao), lif@rpi.edu (F. Li), jxqiu@xmu.edu.cn (J. Qiu).

http://dx.doi.org/10.1016/j.jcp.2014.10.027
0021-9991/© 2014 Elsevier Inc. All rights reserved.

with suitable initial and boundary conditions. Here (1.1) can be scalar or a system, and it is often nonlinear. Our methods use Hermite WENO (HWENO) reconstructions as spatial discretizations, and Lax–Wendroff type discretizations or the natural continuous extension of Runge–Kutta methods as time discretizations, in a central finite volume formulation on staggered meshes. Compared with central WENO (C-WENO) schemes, one major advantage of C-HWENO schemes is the compactness in the spatial reconstruction. Compared with the original HWENO schemes, the proposed methods require neither flux splitting nor the use of numerical fluxes that are often exact or approximate Riemann solvers. When (1.1) is a system, local characteristic decomposition is applied in the reconstruction of cell averages to enhance non-oscillatory property of the schemes.

WENO schemes are high-order finite volume or finite difference methods widely used for hyperbolic conservation laws, with attractive property of maintaining both uniform high-order accuracy and an essentially non-oscillatory shock transition. They were designed based on the successful ENO (essentially non-oscillatory) schemes [7,29,30], and improve in robustness, smoothness of fluxes, better steady state convergence, accuracy in smooth region of the solutions, and efficiency. The first WENO scheme was constructed in [18] as a third-order finite volume method in one space dimension. In [9], third and fifth-order finite difference WENO schemes in multiple dimensions were constructed, with a general framework to design smoothness indicators and nonlinear weights. Finite difference WENO schemes of higher order accuracy (i.e. seventh- to eleventh-order) were proposed in [2], while the finite volume versions on structured and unstructured meshes were investigated in, e.g. [5,8,14,27,23]. In [27], a simple and effective technique for handling negative linear weights without a need to get rid of them was proposed, and this technique is also adopted in this paper. We refer to [28] for a detailed review of WENO schemes. It is known that higher order accuracy in a finite difference or finite volume framework relies on enlarging the stencil for reconstructions. To improve the compactness while keeping the accurate and non-oscillatory properties of the methods, a fifth-order finite volume HWENO scheme was proposed in [24] for one dimension, with a fourth-order one in [26,34] for two dimensions. The compactness of HWENO methods is achieved by evolving not only the solution but also its first spatial derivative(s), and they are both used in the high-order spatial reconstructions. Some other related earlier work includes [31,4,21].

Our proposed methods are also related to central schemes, which can be regarded as extensions of the classical Lax–Friedrichs method [6]. A second-order central scheme was first developed in [22] by Nessyahu and Tadmor, and it requires neither numerical fluxes, that are exact or approximate Riemann solvers, nor flux splitting. For the system case, numerical tests show that local characteristic decomposition is also not needed. Motivated by the simplicity and robustness of the second-order central scheme, various high-order or semi-discrete versions as well as extensions to multiple dimensions were explored in [19,3,1,10–12,14,15,23]. In a series of recent papers, the ENO and WENO reconstruction techniques have been successfully integrated into the central framework. A one-dimensional central ENO (C-ENO) scheme was introduced in [3]. The third- and fourth-order C-WENO schemes were developed in [14–17] for one- and two-dimensional conservation laws. In [23], fifth- and ninth-order C-WENO schemes were constructed based on finite volume formulation on staggered meshes, and they used the natural continuous extension of Runge–Kutta methods in time. Numerical experiments in [23] also demonstrate that the local characteristic decomposition is still necessary to control spurious oscillations when the order of accuracy is high, for both the central WENO schemes on staggered meshes and the upwind WENO schemes on non-staggered meshes.

When upwind type WENO schemes are used as spatial discretizations for solving hyperbolic conservation laws, they are often combined with explicit nonlinearly stable TVD Runge–Kutta time discretizations [29] following method of lines approaches. The schemes developed in the present work, on the other hand, are defined on staggered meshes, and they are more of fully discrete schemes themselves. For such methods, one can no longer directly apply the explicit nonlinearly stable TVD Runge–Kutta methods to achieve high-order accuracy in time. Instead, we choose to use two other time discretizations: the Lax–Wendroff type discretizations, and the natural continuous extension of Runge–Kutta methods. The one-step one-stage Lax–Wendroff type time discretization, which is also called the Taylor type, is based on the idea of the classical Lax–Wendroff scheme [13]. It relies on converting the time derivatives in a temporal Taylor expansion of the solution into spatial derivatives by repeatedly using the governing equations and their differentiated forms. The original finite volume ENO schemes in [7] used this approach for the time discretization. In [25], a Lax–Wendroff time discretization procedure was developed for high-order finite difference WENO schemes. In contrast to upwind schemes, many fully discrete high-order central schemes [3,14–17,23] use Runge–Kutta methods with the aid of the natural continuous extension [33] as time discretizations. The natural continuous extension of Runge–Kutta technique is based on standard Runge–Kutta time discretizations, and it provides approximations of comparable accuracy for the solutions at intermediate time over each time step. It works well with spatial discretizations defined on staggered meshes yet does not require much additional cost than standard Runge–Kutta methods.

The organization of this paper is as follows. In Section 2, we describe in detail the construction and implementation of C-HWENO schemes with Lax–Wendroff type time discretizations for one- and two-dimensional scalar and system equation (1.1). In Section 3, we present the construction and implementation of C-HWENO schemes with the natural continuous extension of Runge–Kutta methods as time discretizations, and the HWENO reconstructions here are similar to those in Section 2. In Section 4, extensive numerical examples are provided to demonstrate the performance of the proposed schemes for smooth and non-smooth examples in one and two dimensions. Concluding remarks are made in Section 5.

2. Central Hermite WENO scheme with Lax–Wendroff time discretization

In this section we describe in detail the construction and implementation of the central HWENO schemes with Lax–Wendroff time discretizations for one- and two-dimensional scalar and system conservation laws. In particular, a fifth-order spatial reconstruction is formulated for one dimension, and a fourth-order one is for two dimensions.

2.1. One-dimensional case

Consider the one-dimensional scalar conservation law

$$\begin{cases} u_t + f(u)_x = 0, \\ u(x,0) = u_0(x). \end{cases} \tag{2.1}$$

Letting $v = u_x$, and taking spatial derivative of (2.1), we obtain

$$\begin{cases} v_t + \big(f'(u)v\big)_x = 0, \\ v(x,0) = \big(u_0(x)\big)_x. \end{cases} \tag{2.2}$$

The proposed numerical method will be formulated based on (2.1) and (2.2), and defined on staggered meshes. For simplicity of presentation, uniform meshes are used with the mesh size Δx. Each cell of the *primal* mesh is denoted as $I_i = [x_{i-1/2}, x_{i+1/2}]$ with its cell center $x_i = \frac{1}{2}(x_{i-1/2}+x_{i+1/2})$; and each cell of the *dual* mesh is denoted as $I_{i+1/2} = [x_i, x_{i+1}]$ with its cell center $x_{i+1/2} = \frac{1}{2}(x_i + x_{i+1})$. It will be seen that the primal and the dual meshes will be used in a staggered fashion along the time direction in the proposed schemes.

Our method starts with a Lax–Wendroff type temporal discretization. Let Δt denote a time step. By a temporal Taylor expansion we obtain

$$q(x, t+\Delta t) = q(x,t) + \Delta t q_t + \frac{\Delta t^2}{2} q_{tt} + \frac{\Delta t^3}{6} q_{ttt} + \cdots, \quad q = u \text{ or } v. \tag{2.3}$$

If we want to obtain the kth-order accuracy in time, we need to approximate the first k time derivatives: $\frac{\partial q}{\partial t}, \cdots, \frac{\partial^{(k)} q}{\partial t^k}$, $q = u$ or v. In this paper, we will proceed up to the third-order accuracy in time, although the procedure can be extended directly to any other order.

Next we convert the temporal derivative terms in (2.3) into spatial ones by repeatedly using the governing equations (2.1) and (2.2):

$$\begin{aligned}
&u_t = -f(u)_x = -f'(u)v, \qquad u_{tt} = -\big(f'(u)u_t\big)_x = -f''(u)vu_t - f'(u)u_{xt}, \\
&u_{xt} = -f''(u)(v)^2 - f'(u)v_x, \qquad u_{ttt} = -\big(f''(u)(u_t)^2 + f'(u)u_{tt}\big)_x, \\
&v_t = -\big(f'(u)v\big)_x = -f''(u)(v)^2 - f'(u)v_x, \\
&v_{tt} = -\big(f''(u)u_t v + f'(u)v_t\big)_x = -\big(f'''(u)vu_t v + f''(u)(u_{xt}v + u_t v_x + vv_t) + f'(u)v_{xt}\big), \\
&v_{xt} = -\big(f''(u)(v)^2 + f'(u)v_x\big)_x = -\big(f'''(u)(v)^3 + 3f''(u)vv_x + f'(u)v_{xx}\big), \\
&v_{ttt} = -\big(\big(f''(u)u_t v + f'(u)v_t\big)_t\big)_x = -\big(f'''(u)(u_t)^2 v + f''(u)(u_{tt}v + 2u_t v_t) + f'(u)v_{tt}\big)_x.
\end{aligned} \tag{2.4}$$

A third-order approximation of (2.3) can be rewritten as

$$\begin{aligned}
&u(x, t+\Delta t) \approx u(x,t) - \Delta t F_x, \\
&v(x, t+\Delta t) \approx v(x,t) - \Delta t G_x,
\end{aligned} \tag{2.5}$$

with

$$\begin{aligned}
&F = f + \frac{\Delta t}{2} f'(u)u_t + \frac{\Delta t^2}{6}\big(f''(u)(u_t)^2 + f'(u)u_{tt}\big), \\
&G = f'(u)v + \frac{\Delta t}{2}\big(f''(u)u_t v + f'(u)v_t\big) + \frac{\Delta t^2}{6}\big(f'''(u)(u_t)^2 v + f''(u)(u_{tt}v + 2u_t v_t) + f'(u)v_{tt}\big),
\end{aligned} \tag{2.6}$$

where all the time derivatives in (2.6) are replaced by spatial derivatives according to the relations in (2.4).

Now we apply the staggered central scheme strategy to (2.5). Suppose at $t = t^n$, the approximations for the cell averages of the solution and its first derivative, denoted as $\{\bar{u}_i^n\}$ and $\{\bar{v}_i^n\}$, are available on the primal mesh, that is, $\forall i$,

$$\bar{u}_i^n \approx \frac{1}{\Delta x}\int_{I_i} u\big(x, t^n\big)dx, \qquad \bar{v}_i^n \approx \frac{1}{\Delta x}\int_{I_i} v\big(x, t^n\big)dx.$$

(a.1) We integrate (2.5) with $t = t^n$ over $I_{i+1/2}$, and approximate the cell averages of the solution and its first derivative with respect to the dual mesh at $t^{n+1} = t^n + \Delta t$ as below,

$$\bar{u}^{n+1}_{i+1/2} = \bar{u}^n_{i+1/2} - \frac{\Delta t}{\Delta x}\big[F\big(u(x_{i+1}, t^n)\big) - F\big(u(x_i, t^n)\big)\big], \tag{2.7}$$

$$\bar{v}^{n+1}_{i+1/2} = \bar{v}^n_{i+1/2} - \frac{\Delta t}{\Delta x}\big[G\big(u(x_{i+1}, t^n), v(x_{i+1}, t^n)\big) - G\big(u(x_i, t^n), v(x_i, t^n)\big)\big]. \tag{2.8}$$

(a.2) We then integrate (2.5) with $t = t^{n+1}$ over I_i, and continue to approximate the cell averages of the solution and its first derivative with respect to the primal mesh at $t^{n+2} = t^{n+1} + \Delta t$,

$$\bar{u}^{n+2}_i = \bar{u}^{n+1}_i - \frac{\Delta t}{\Delta x}\big[F\big(u(x_{i+1/2}, t^{n+1})\big) - F\big(u(x_{i-1/2}, t^{n+1})\big)\big],$$

$$\bar{v}^{n+2}_i = \bar{v}^{n+1}_i - \frac{\Delta t}{\Delta x}\big[G\big(u(x_{i+1/2}, t^{n+1}), v(x_{i+1/2}, t^{n+1})\big) - G\big(u(x_{i-1/2}, t^{n+1}), v(x_{i-1/2}, t^{n+1})\big)\big].$$

(a.3) Set n to be $n + 2$, and go to (a.1).

Note that the cell averages are defined and evolved in a staggered fashion with respect to the discrete time level n on two sets of meshes. The mesh switches back after two time steps.

The remaining of this section will be mainly devoted to the details to update from t^n to t^{n+1} according to (2.7) and (2.8). Specifically, to obtain the cell averages $\bar{u}^{n+1}_{i+1/2}$ and $\bar{v}^{n+1}_{i+1/2}$ on the dual mesh at the next time t^{n+1} according to (2.7) and (2.8), one will need to reconstruct at the current time t^n, $\forall i$,

(1) the cell averages $\bar{u}^n_{i+1/2}$, $\bar{v}^n_{i+1/2}$ on the dual mesh, as well as
(2) the point value of $q(x_i, t^n)$ at the mid-point x_i of I_i, where $q = u, v, v_x$, or v_{xx},

based on the given data $\{\bar{u}^n_i, \bar{v}^n_i\}_i$. Recall that F and G are functions depending on u and the derivatives v, v_x, v_{xx}. On the other hand,

$$\bar{u}^n_{i+1/2} \approx \frac{1}{\Delta x}\int_{x_i}^{x_{i+1}} u(x, t^n)dx = \frac{1}{\Delta x}\int_{x_i}^{x_{i+1/2}} u(x, t^n)dx + \frac{1}{\Delta x}\int_{x_{i+1/2}}^{x_{i+1}} u(x, t^n)dx.$$

This implies that to approximate the cell average $\bar{u}^n_{i+1/2}$, one would want to get the half cell averages $\frac{1}{\Delta x}\int_{x_i}^{x_{i+1/2}} u(x, t^n)dx$ and $\frac{1}{\Delta x}\int_{x_{i+1/2}}^{x_{i+1}} u(x, t^n)dx$. This is likewise for $\bar{v}^n_{i+1/2}$.

To reconstruct the half cell averages and point values mentioned above, we adapt the one-dimensional fifth-order accurate HWENO reconstruction of Qiu and Shu [24], where only the function values at the end points of each cell were reconstructed. This reconstruction is not only high-order accurate but also essentially non-oscillatory. Compared with the standard WENO schemes, the stencils in the reconstruction are more compact. Next in steps 1–6, we will describe the reconstruction with great details, which is based on the cell averages $\{\bar{u}^n_i, \bar{v}^n_i\}_i$ on the primal mesh at $t = t^n$. One will see that the reconstructions discussed in steps 1–5 are HWENO-type, yet the one in step 6 is linear. Since the data before and after the reconstruction is all at the same time level t^n, the superscript n will be omitted below, together with the dependence on the time t of u and v.

Step 1. A HWENO reconstruction of $\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} u(x)dx$ from the cell averages $\{\bar{u}_i, \bar{v}_i\}_i$.

We first introduce three "small" stencils $S_0 = \{I_{i-1}, I_i\}$, $S_1 = \{I_i, I_{i+1}\}$, $S_2 = \{I_{i-1}, I_i, I_{i+1}\}$, one "large" stencil $\mathcal{T} = \{S_0, S_1, S_2\} = S_2$, and reconstruct three quadratic Hermite polynomials $p_0(x)$, $p_1(x)$, $p_2(x)$ on S_0, S_1, S_2, respectively, and one quartic Hermite polynomial $Q(x)$ on $\mathcal{T}$, satisfying the following conditions,

$$\begin{aligned}
&\frac{1}{\Delta x}\int_{I_{i+j}} p_0(x)dx = \bar{u}_{i+j}, \quad j = -1, 0, && \frac{1}{\Delta x}\int_{I_{i-1}} p_0'(x)dx = \bar{v}_{i-1},\\
&\frac{1}{\Delta x}\int_{I_{i+j}} p_1(x)dx = \bar{u}_{i+j}, \quad j = 0, 1, && \frac{1}{\Delta x}\int_{I_{i+1}} p_1'(x)dx = \bar{v}_{i+1},\\
&\frac{1}{\Delta x}\int_{I_{i+j}} p_2(x)dx = \bar{u}_{i+j}, \quad j = -1, 0, 1, &&\\
&\frac{1}{\Delta x}\int_{I_{i+j}} Q(x)dx = \bar{u}_{i+j}, \quad j = -1, 0, 1, && \frac{1}{\Delta x}\int_{I_{i+j}} Q'(x)dx = \bar{v}_{i+j}, \quad j = -1, 1.
\end{aligned} \tag{2.9}$$

To reconstruct $\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} u(x)dx$, we further compute the half cell averages of these polynomials over $[x_{i-1/2}, x_i]$, and the results can be given explicitly in terms of the cell averages $\{\bar{u}_i, \bar{v}_i\}_i$,

$$\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} p_0(x)dx = \frac{1}{4}\bar{u}_{i-1} + \frac{1}{4}\bar{u}_i + \frac{\Delta x}{8}\bar{v}_{i-1},$$

$$\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} p_1(x)dx = \frac{3}{4}\bar{u}_i - \frac{1}{4}\bar{u}_{i+1} + \frac{\Delta x}{8}\bar{v}_{i+1},$$

$$\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} p_2(x)dx = \frac{1}{16}\bar{u}_{i-1} + \frac{1}{2}\bar{u}_i - \frac{1}{16}\bar{u}_{i+1},$$

$$\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} Q(x)dx = \frac{25}{256}\bar{u}_{i-1} + \frac{1}{2}\bar{u}_i - \frac{25}{256}\bar{u}_{i+1} + \frac{9\Delta x}{256}\bar{v}_{i-1} + \frac{9\Delta x}{256}\bar{v}_{i+1}. \tag{2.10}$$

Now one can find the linear weights (or combination coefficients), denoted as γ_0, γ_1 and γ_2, such that

$$\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} Q(x)dx = \frac{1}{\Delta x}\sum_{j=0}^{2}\gamma_j\int_{x_{i-1/2}}^{x_i} p_j(x)dx$$

for all possible values of the cell averages of u and v over the stencil $\mathcal{T}$. This requirement leads to

$$\gamma_0 = \frac{9}{32}, \qquad \gamma_1 = \frac{9}{32}, \qquad \gamma_2 = \frac{14}{32}. \tag{2.11}$$

With the linear weights, we combine the relatively lower order approximations, which are third-order accurate here, into a higher order approximation that is fifth-order accurate in our case.

An important ingredient of WENO-type methods for solving hyperbolic conservation laws with strong shocks or other discontinuities in the solutions is nonlinear weights, which are applied to control spurious oscillations. To obtain the nonlinear weights, we compute a smoothness indicator β_j for each stencil S_j $(j = 0, 1, 2)$, which measures how smooth the function $p_j(x)$ is in the target cell I_i. The smaller the smoothness indicator β_j is, the smoother the function $p_j(x)$ is in the cell I_i. Following [9], the smoothness indicator β_j is defined as below,

$$\beta_j = \sum_{l=1}^{2}\int_{I_i}\Delta x^{2l-1}\left(\frac{\partial^l}{\partial x^l}p_j(x)\right)^2 dx, \tag{2.12}$$

and it can be further given explicitly in terms of the cell averages of u and v for the convenience of the actual implementation,

$$\begin{cases} \beta_0 = (-2\bar{u}_{i-1} + 2\bar{u}_i - \Delta x\bar{v}_{i-1})^2 + \frac{13}{3}(-\bar{u}_{i-1} + \bar{u}_i - \Delta x\bar{v}_{i-1})^2, \\ \beta_1 = (-2\bar{u}_i + 2\bar{u}_{i+1} - \Delta x\bar{v}_{i+1})^2 + \frac{13}{3}(-\bar{u}_i + \bar{u}_{i+1} - \Delta x\bar{v}_{i+1})^2, \\ \beta_2 = \frac{1}{4}(-\bar{u}_{i-1} + \bar{u}_{i+1})^2 + \frac{13}{12}(-\bar{u}_{i-1} + 2\bar{u}_i - \bar{u}_{i+1})^2. \end{cases} \tag{2.13}$$

With the smoothness indicators $\{\beta_j\}_j$ in (2.13) and the linear weights $\{\gamma_j\}_j$ in (2.11), we can now compute the nonlinear weights ω_j, $j = 0, 1, 2$,

$$\omega_j = \frac{\bar{\omega}_j}{\sum_{k=0}^{2}\bar{\omega}_k}, \quad \text{where } \bar{\omega}_k = \frac{\gamma_k}{(\varepsilon + \beta_k)^2}, \; k = 0, 1, 2.$$

Here $\varepsilon > 0$ is a small constant to avoid the denominator to be zero. We use $\varepsilon = 10^{-6}$ in all numerical examples in this paper.

Finally, a fifth-order HWENO approximation for $\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} u(x)dx$ is given as

$$\frac{1}{\Delta x}\int\limits_{x_{i-1/2}}^{x_i} u(x)dx \approx \frac{1}{\Delta x}\sum_{j=0}^{2}\omega_j \int\limits_{x_{i-1/2}}^{x_i} p_j(x)dx,$$

where $\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} p_j(x)dx$, $j=0,1,2$ are given in (2.10).

In the target cell I_i, the right half cell average of u over $[x_i, x_{i+1/2}]$ can be approximated based on the local conservation of u,

$$\frac{1}{\Delta x}\int\limits_{x_i}^{x_{i+1/2}} u(x)dx \approx \bar{u}_i - \frac{1}{\Delta x}\int\limits_{x_{i-1/2}}^{x_i} u(x)dx.$$

Step 2. A HWENO reconstruction of $u(x_i)$ from the cell averages $\{\bar{u}_i, \bar{v}_i\}_i$.

In this step, the same stencils S_0, S_1, S_2, $\mathcal{T}$ as in step 1 are used, together with the same constructed polynomials $p_0(x)$, $p_1(x)$, $p_2(x)$ and $Q(x)$ in (2.9). The point value of these polynomials at x_i can be expressed in terms of the cell averages,

$$\begin{aligned}
p_0(x_i) &= \frac{1}{12}\bar{u}_{i-1} + \frac{11}{12}\bar{u}_i + \frac{\Delta x}{12}\bar{v}_{i-1},\\
p_1(x_i) &= \frac{11}{12}\bar{u}_i + \frac{1}{12}\bar{u}_{i+1} - \frac{\Delta x}{12}\bar{v}_{i+1},\\
p_2(x_i) &= -\frac{1}{24}\bar{u}_{i-1} + \frac{13}{12}\bar{u}_i - \frac{1}{24}\bar{u}_{i+1},\\
Q(x_i) &= -\frac{47}{480}\bar{u}_{i-1} + \frac{287}{240}\bar{u}_i - \frac{47}{480}\bar{u}_{i+1} - \frac{9\Delta x}{320}\bar{v}_{i-1} + \frac{9\Delta x}{320}\bar{v}_{i+1}.
\end{aligned}$$

The linear weights γ_0, γ_1, γ_2 are chosen such that

$$Q(x_i) = \sum_{j=0}^{2}\gamma_j p_j(x_i)$$

for all possible values of the cell averages of u and v over the stencil $\mathcal{T}$, and this requirement leads to

$$\gamma_0 = -\frac{27}{80}, \qquad \gamma_1 = -\frac{27}{80}, \qquad \gamma_2 = \frac{134}{80}. \tag{2.14}$$

Note that two linear weights are negative in (2.14), and the corresponding WENO-type approximation with such weights can be unstable. To avoid this, we follow the splitting technique developed in [27] to handle negative weights in WENO schemes. More specifically, we split the linear weights into two groups

$$\tilde{\gamma}_j^+ = \frac{1}{2}\left(\gamma_j + 3|\gamma_j|\right), \qquad \tilde{\gamma}_j^- = \tilde{\gamma}_j^+ - \gamma_j, \quad j=0,1,2,$$

which can be shown to satisfy

$$\tilde{\gamma}_j^+ = 2\gamma_j, \qquad \tilde{\gamma}_j^- = \gamma_j, \quad \text{for } \gamma_j > 0; \qquad \tilde{\gamma}_j^+ = -\gamma_j, \qquad \tilde{\gamma}_j^- = -2\gamma_j, \quad \text{for } \gamma_j < 0.$$

We further scale the terms by introducing

$$\sigma^\pm = \sum_{j=0}^{2}\tilde{\gamma}_j^\pm, \qquad \gamma_j^\pm = \tilde{\gamma}_j^\pm/\sigma^\pm, \quad j=0,1,2.$$

For the linear weights in (2.14), this gives

$$\begin{aligned}
&\sigma^+ = \frac{161}{40}, \qquad \sigma^- = \frac{242}{80},\\
&\gamma_0^+ = \frac{27}{322}, \qquad \gamma_1^+ = \frac{27}{322}, \qquad \gamma_2^+ = \frac{134}{161}, \qquad \gamma_0^- = \frac{27}{121}, \qquad \gamma_1^- = \frac{27}{121}, \qquad \gamma_2^- = \frac{134}{242}.
\end{aligned}$$

The HWENO reconstruction will be performed for each group by computing the nonlinear weights ω_j^+ and ω_j^-, $j=0,1,2$ separately, with the same smoothness indicators $\{\beta_j\}_j$ as in (2.13),

$$\omega_j^\pm = \frac{\bar{\omega}_j^\pm}{\sum_{k=0}^{2}\bar{\omega}_k^\pm}, \quad \text{where } \bar{\omega}_k^\pm = \frac{\gamma_k^\pm}{(\varepsilon + \beta_k)^2}, \; k=0,1,2.$$

The final HWENO reconstruction of $u(x_i)$ is now taken as a combination of the reconstructions using the two groups of weights

$$u(x_i) \approx \sigma^+ \sum_{j=0}^{2} \omega_j^+ p_j(x_i) - \sigma^- \sum_{j=0}^{2} \omega_j^- p_j(x_i).$$

In the adopted technique to treat negative linear weights, the key is to ensure that every stencil has a significant representation in both positive and negative weight groups. Within each group, one still follows the standard HWENO idea of redistributing the weights subject to a fixed sum according to the smoothness of the approximations.

Step 3. A HWENO reconstruction of $\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} v(x)dx$ from the cell averages $\{\bar{u}_i, \bar{v}_i\}_i$.

In this step, the same stencils S_0, S_1, S_2, $\mathcal{T}$ as in step 1 are used. We reconstruct three cubic Hermite polynomials $p_0(x)$, $p_1(x)$, $p_2(x)$ on S_0, S_1, S_2, respectively, and one quintic Hermite polynomial $Q(x)$ on $\mathcal{T}$, satisfying

$$\begin{aligned}
&\frac{1}{\Delta x}\int_{I_{i+j}} p_0(x)dx = \bar{u}_{i+j}, \qquad \frac{1}{\Delta x}\int_{I_{i+j}} p_0'(x)dx = \bar{v}_{i+j}, \quad j=-1,0,\\
&\frac{1}{\Delta x}\int_{I_{i+j}} p_1(x)dx = \bar{u}_{i+j}, \qquad \frac{1}{\Delta x}\int_{I_{i+j}} p_1'(x)dx = \bar{v}_{i+j}, \quad j=0,1,\\
&\frac{1}{\Delta x}\int_{I_{i+j}} p_2(x)dx = \bar{u}_{i+j}, \quad j=-1,0,1, \qquad \frac{1}{\Delta x}\int_{I_i} p_2'(x)dx = \bar{v}_i,\\
&\frac{1}{\Delta x}\int_{I_{i+j}} Q(x)dx = \bar{u}_{i+j}, \qquad \frac{1}{\Delta x}\int_{I_{i+j}} Q'(x)dx = \bar{v}_{i+j}, \quad j=-1,0,1.
\end{aligned}$$

To reconstruct $\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} v(x)dx$, we further compute the half cell averages of the first derivative of these polynomials over $[x_{i-1/2}, x_i]$, and the results can be given explicitly in terms of the cell averages $\{\bar{u}_i, \bar{v}_i\}_i$,

$$\begin{aligned}
&\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} p_0'(x)dx = -\frac{3}{4\Delta x}\bar{u}_{i-1} + \frac{3}{4\Delta x}\bar{u}_i - \frac{1}{4}\bar{v}_{i-1},\\
&\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} p_1'(x)dx = \frac{3}{4\Delta x}\bar{u}_i - \frac{3}{4\Delta x}\bar{u}_{i+1} + \bar{v}_i + \frac{1}{4}\bar{v}_{i+1},\\
&\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} p_2'(x)dx = -\frac{1}{8\Delta x}\bar{u}_{i-1} + \frac{1}{4\Delta x}\bar{u}_i - \frac{1}{8\Delta x}\bar{u}_{i+1} + \frac{1}{2}\bar{v}_i,\\
&\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} Q'(x)dx = -\frac{9}{32\Delta x}\bar{u}_{i-1} + \frac{9}{16\Delta x}\bar{u}_i - \frac{9}{32\Delta x}\bar{u}_{i+1} - \frac{5}{64}\bar{v}_{i-1} + \frac{1}{2}\bar{v}_i + \frac{5}{64}\bar{v}_{i+1}.
\end{aligned} \tag{2.15}$$

Just as in step 1, we next want to obtain the nonlinear weights ω_j, $j=0,1,2$, with which, a fifth-order HWENO reconstruction can be given for $\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} v(x)dx$. To achieve this, we first find the linear weights, denoted as γ_0', γ_1' and γ_2', such that

$$\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} Q'(x)dx = \frac{1}{\Delta x}\sum_{j=0}^{2}\gamma_j' \int_{x_{i-1/2}}^{x_i} p_j'(x)dx$$

for all possible values of the cell averages of u and v over the stencil $\mathcal{T}$, and this leads to

$$\gamma_0' = \frac{5}{16}, \qquad \gamma_1' = \frac{5}{16}, \qquad \gamma_2' = \frac{6}{16}. \tag{2.16}$$

We then compute the smoothness indicators β_j, $j=0,1,2$ as below

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

Since we are reconstructing the first derivative rather than the solution itself, the summation above starts from the second derivative to the rth derivative, where r is the degree of the polynomial $p_j(x)$. For the convenience of the actual implementation, the smoothness indicators $\{\beta_j\}_{j=0}^2$ can be further given in quadratic forms of the cell averages of u and v,

$$\begin{cases} \beta_0 = 4\big(3(\bar{u}_{i-1} - \bar{u}_i) + \Delta x(\bar{v}_{i-1} + 2\bar{v}_i)\big)^2 + 39\big(2(\bar{u}_{i-1} - \bar{u}_i) + \Delta x(\bar{v}_{i-1} + \bar{v}_i)\big)^2, \\ \beta_1 = 4\big(3(\bar{u}_i - \bar{u}_{i+1}) + \Delta x(2\bar{v}_i + \bar{v}_{i+1})\big)^2 + 39\big(2(\bar{u}_i - \bar{u}_{i+1}) + \Delta x(\bar{v}_i + \bar{v}_{i+1})\big)^2, \\ \beta_2 = (\bar{u}_{i-1} - 2\bar{u}_i + \bar{u}_{i+1})^2 + \dfrac{39}{4}(\bar{u}_{i+1} - \bar{u}_{i-1} - 2\Delta x\bar{v}_i)^2. \end{cases} \tag{2.17}$$

With the smoothness indicators $\{\beta_j\}_j$ in (2.17) and the linear weights $\{\gamma_j'\}_j$ in (2.16), the nonlinear weights ω_j, $j = 0, 1, 2$, can be computed,

$$\omega_j = \frac{\bar{\omega}_j}{\sum_{k=0}^2 \bar{\omega}_k}, \quad \text{where } \bar{\omega}_k = \frac{\gamma_k'}{(\varepsilon + \beta_k)^2}, \ k = 0, 1, 2.$$

Finally, a fifth-order HWENO approximation for $\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} v(x)dx$ is given as

$$\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} v(x)dx \approx \frac{1}{\Delta x}\sum_{j=0}^{2}\omega_j \int_{x_{i-1/2}}^{x_i} p_j'(x)dx.$$

In the target cell I_i, the right half cell average of v over $[x_i, x_{i+1/2}]$ can be approximated based on the local conservation of v,

$$\frac{1}{\Delta x}\int_{x_i}^{x_{i+1/2}} v(x)dx \approx \bar{v}_i - \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i} v(x)dx.$$

Step 4. A HWENO reconstruction of $v(x_i)$ from the cell averages $\{\bar{u}_i, \bar{v}_i\}_i$.

In this step, we use the same stencils S_0, S_1, S_2, $\mathcal{T}$ and the same constructed polynomials $p_0(x)$, $p_1(x)$, $p_2(x)$, $Q(x)$ as in step 3. The point values of the first derivative of these polynomials at x_i can be expressed in terms of the cell averages,

$$\begin{aligned} p_0'(x_i) &= -\frac{1}{2\Delta x}\bar{u}_{i-1} + \frac{1}{2\Delta x}\bar{u}_i - \frac{1}{4}\bar{v}_{i-1} + \frac{3}{4}\bar{v}_i, \\ p_1'(x_i) &= -\frac{1}{2\Delta x}\bar{u}_i + \frac{1}{2\Delta x}\bar{u}_{i+1} + \frac{3}{4}\bar{v}_i - \frac{1}{4}\bar{v}_{i+1}, \\ p_2'(x_i) &= \frac{1}{8\Delta x}\bar{u}_{i-1} - \frac{1}{8\Delta x}\bar{u}_{i+1} + \frac{5}{4}\bar{v}_i, \\ Q'(x_i) &= \frac{27}{64\Delta x}\bar{u}_{i-1} - \frac{27}{64\Delta x}\bar{u}_{i+1} + \frac{19}{192}\bar{v}_{i-1} + \frac{79}{48}\bar{v}_i + \frac{19}{192}\bar{v}_{i+1}. \end{aligned}$$

The linear weights γ_0', γ_1', γ_2' can be determined by requiring

$$Q'(x_i) = \sum_{j=0}^{2}\gamma_j' p_j'(x_i)$$

for all possible values of the cell averages of u and v over the stencil $\mathcal{T}$, and this leads to

$$\gamma_0' = -\frac{19}{48}, \qquad \gamma_1' = -\frac{19}{48}, \qquad \gamma_2' = \frac{86}{48}. \tag{2.18}$$

There are two negative linear weights in (2.18). Following the procedure provided in step 2, we get

$$\sigma'^{+} = \frac{105}{24}, \qquad \sigma'^{-} = \frac{162}{48},$$

$$\gamma_0'^{+} = \frac{19}{210}, \qquad \gamma_1'^{+} = \frac{19}{210}, \qquad \gamma_2'^{+} = \frac{86}{105}, \qquad \gamma_0'^{-} = \frac{38}{162}, \qquad \gamma_1'^{-} = \frac{38}{162}, \qquad \gamma_2'^{-} = \frac{86}{162}.$$

The HWENO reconstruction will be performed for each group by computing the nonlinear weights ω_j^{+} and ω_j^{-}, $j = 0, 1, 2$ separately, with the same smoothness indicators $\{\beta_j\}_j$ as in (2.17),

$$\omega_j^{\pm} = \frac{\bar{\omega}_j^{\pm}}{\sum_{k=0}^2 \bar{\omega}_k^{\pm}}, \quad \text{where } \bar{\omega}_k^{\pm} = \frac{\gamma'^{\pm}_k}{(\varepsilon + \beta_k)^2}, \ k = 0, 1, 2.$$

The final HWENO reconstruction of $v(x_i)$ is now obtained as

$$v(x_i) \approx \sigma'^{+} \sum_{j=0}^{2} \omega_j^{+} p_j'(x_i) - \sigma'^{-} \sum_{j=0}^{2} \omega_j^{-} p_j'(x_i).$$

Step 5. A HWENO reconstruction of $v_x(x_i)$ from the cell averages $\{\bar{u}_i, \bar{v}_i\}_i$.

In this step, we use the same stencils S_0, S_1, S_2, $\mathcal{T}$ and the same constructed polynomials $p_0(x)$, $p_1(x)$, $p_2(x)$, $Q(x)$ as in step 3. The point values of the second derivative of these polynomials at x_i can be expressed in terms of the cell averages,

$$\begin{aligned}
p_0''(x_i) &= \frac{6}{\Delta x^2}\bar{u}_{i-1} - \frac{6}{\Delta x^2}\bar{u}_i + \frac{2}{\Delta x}\bar{v}_{i-1} + \frac{4}{\Delta x}\bar{v}_i, \\
p_1''(x_i) &= -\frac{6}{\Delta x^2}\bar{u}_i + \frac{6}{\Delta x^2}\bar{u}_{i+1} - \frac{4}{\Delta x}\bar{v}_i - \frac{2}{\Delta x}\bar{v}_{i+1}, \\
p_2''(x_i) &= \frac{1}{\Delta x^2}\bar{u}_{i-1} - \frac{2}{\Delta x^2}\bar{u}_i + \frac{1}{\Delta x^2}\bar{u}_{i+1}, \\
Q''(x_i) &= \frac{5}{2\Delta x^2}\bar{u}_{i-1} - \frac{5}{\Delta x^2}\bar{u}_i + \frac{5}{2\Delta x^2}\bar{u}_{i+1} + \frac{3}{4\Delta x}\bar{v}_{i-1} - \frac{3}{4\Delta x}\bar{v}_{i+1}.
\end{aligned}$$

The linear weights γ_0'', γ_1'', γ_2'' are determined by requiring

$$Q''(x_i) = \sum_{j=0}^{2} \gamma_j'' p_j''(x_i)$$

for all possible values of the cell averages of u and v over the stencil $\mathcal{T}$, and this leads to

$$\gamma_0'' = \frac{3}{8}, \qquad \gamma_1'' = \frac{3}{8}, \qquad \gamma_2'' = \frac{2}{8}. \tag{2.19}$$

We then compute the smoothness indicators β_j, $j = 0, 1, 2$ as below

$$\beta_j = \Delta x^5 \int_{I_i} \left(\frac{\partial^3}{\partial x^3} p_j(x) \right)^2 dx.$$

Since we are reconstructing the second derivative rather than the solution itself, the smoothness indicator β_j only depends on the third derivative of the cubic polynomial $p_j(x)$. For the convenience of the actual implementation, the smoothness indicators $\{\beta_j\}_{j=0}^2$ can be further given in terms of the cell averages of u and v,

$$\begin{cases}
\beta_0 = 36\big(2(\bar{u}_{i-1} - \bar{u}_i) + \Delta x(\bar{v}_{i-1} + \bar{v}_i)\big)^2, \\
\beta_1 = 36\big(2(\bar{u}_i - \bar{u}_{i+1}) + \Delta x(\bar{v}_i + \bar{v}_{i+1})\big)^2, \\
\beta_2 = 9(\bar{u}_{i+1} - \bar{u}_{i-1} - 2\Delta x \bar{v}_i)^2.
\end{cases} \tag{2.20}$$

With the smoothness indicators $\{\beta_j\}_j$ in (2.20) and the linear weights in (2.19), we can compute the nonlinear weights ω_j, $j = 0, 1, 2$,

$$\omega_j = \frac{\bar{\omega}_j}{\sum_{k=0}^{2} \bar{\omega}_k}, \quad \text{where } \bar{\omega}_k = \frac{\gamma_k''}{(\varepsilon + \beta_k)^2}, \ k = 0, 1, 2.$$

The final HWENO reconstruction for $v_x(x_i)$ is now obtained as

$$v_x(x_i) \approx \sum_{j=0}^{2} \omega_j p_j''(x_i).$$

Step 6. A linear reconstruction of $v_{xx}(x_i)$ from the cell averages $\{\bar{u}_i, \bar{v}_i\}_i$.

We start with the same stencils S_0, S_1, S_2, $\mathcal{T}$ and the same constructed polynomials $p_0(x)$, $p_1(x)$, $p_2(x)$, $Q(x)$ as in step 3. Recall that HWENO reconstructions are based on nonlinear weights which depend on the smoothness indicators. We here want to reconstruct the third derivative of the solution, and this will require computing the fourth-derivative (and above) of the constructed polynomials $p_0(x)$, $p_1(x)$, $p_2(x)$ which are only cubic. Apparently, this will not lead to an effective smoothness indicator. Fortunately, our numerical experiments show that the following linear reconstruction is sufficient to provide an accurate and stable way to approximate $v_{xx}(x_i)$,

$$v_{xx}(x_i) \approx Q'''(x_i) = -\frac{45}{4\Delta x^3}\bar{u}_{i-1} + \frac{45}{4\Delta x^3}\bar{u}_{i+1} - \frac{11}{4\Delta x^2}\bar{v}_{i-1} - \frac{17}{\Delta x^2}\bar{v}_i - \frac{11}{4\Delta x^2}\bar{v}_{i+1}.$$

Here again, $Q(x)$ is from step 3.

Remark 1. Following the current central scheme framework, in Eqs. (2.7)–(2.8) to update $\bar{u}^{n+1}_{i+1/2}$ and $\bar{v}^{n+1}_{i+1/2}$, the flux functions F and G are evaluated at x_i and the time t^n. This requires the reconstructions of the point values of u, v, v_x, v_{xx} at x_i based on cell averages $\{\bar{u}^n_i, \bar{v}^n_i\}_i$ at the same time level. Since all the related reconstructions (see step 2 and steps 4–6) involve stencils including the cell I_i, and x_i is an interior point, these reconstructed point values are naturally single-valued. Therefore unlike in upwind type methods, there is no need to use numerical fluxes for F and G.

Finally in this section, we will briefly discuss the system case. For the systems of conservation laws, such as the Euler equations of gas dynamics, each of the HWENO or linear reconstructions in steps 1–6 could be performed for the unknown solution component by component. Alternatively this can be performed based on the local characteristic decomposition, which often provides better performance of the schemes yet computationally is more costly. It is usually noted in the literature that such local characteristic decomposition is unnecessary for central schemes. However, most of the referred schemes are second-order, or at most third-order accurate. Qiu and Shu [23] pointed out that the WENO schemes involving componentwise reconstructions may become more oscillatory with the increase of the accuracy order, and this was observed for both the central WENO schemes on staggered meshes and standard WENO schemes on non-staggered meshes. In order to effectively control the spurious oscillations and to enhance the numerical stability, we follow the idea in [23]: we apply the local characteristic decomposition during the reconstructions of half cell averages in step 1 and step 3 based on $\{\bar{u}_i, \bar{v}_i\}_i$. All the remaining reconstructions, which are for point values, are still implemented through a componentwise procedure.

From the method formulation point of view, when one converts the second and higher order time derivatives to spatial derivatives in the system case, one needs to work with terms such as $f'(u)$ (a Jacobian matrix), $f''(u)$ (a 3D "matrix", a tensor), $f'''(u)$ (a 4D "matrix", again a tensor). This can become very involved. We have used MATLAB Symbolic Math Toolbox to assist with the derivation of the proposed methods.

2.2. Two-dimensional case

Consider the two-dimensional scalar conservation law

$$\begin{cases} u_t + f(u)_x + g(u)_y = 0, \\ u(x, y, 0) = u_0(x, y). \end{cases} \tag{2.21}$$

Letting $v = u_x$, $w = u_y$, and taking spatial derivatives of (2.21), we obtain

$$\begin{cases} v_t + \big(f'(u)v\big)_x + \big(g'(u)v\big)_y = 0, \\ v(x, y, 0) = \big(u_0(x, y)\big)_x, \end{cases} \tag{2.22}$$

$$\begin{cases} w_t + \big(f'(u)w\big)_x + \big(g'(u)w\big)_y = 0, \\ w(x, y, 0) = \big(u_0(x, y)\big)_y. \end{cases} \tag{2.23}$$

The proposed numerical method will be formulated based on (2.21)–(2.23), and defined on staggered meshes. For simplicity of presentation, uniform meshes are used with the meshsizes Δx in the x direction, and Δy in the y direction. Each cell of the *primal* mesh is denoted as $I_{ij} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$ with its cell center (x_i, y_j); and each cell of the *dual* mesh is denoted as $I_{i+1/2, j+1/2} = [x_i, x_{i+1}] \times [y_j, y_{j+1}]$ with its cell center $(x_{i+1/2}, y_{j+1/2})$.

Our method starts with a Lax–Wendroff type temporal discretization. Let Δt denote a time step. By a temporal Taylor expansion we obtain

$$q(x, y, t + \Delta t) = q(x, y, t) + \Delta t q_t + \frac{\Delta t^2}{2} q_{tt} + \frac{\Delta t^3}{6} q_{ttt} + \cdots, \quad q = u, v, \text{ or } w. \tag{2.24}$$

In order to obtain the kth-order accuracy in time, we need to approximate the first k time derivatives of u, v and w. Just as in one-dimensional case, we here only consider a third-order discretization in time.

We again use the governing equations (2.21)–(2.23) to convert time derivatives of u, v and w into spatial ones, the resulted relations are given in Appendix A. A third-order approximation of (2.24) can be written as

$$\begin{aligned} u(x, y, t + \Delta t) &\approx u(x, y, t) - \Delta t(F_x + G_y), \\ v(x, y, t + \Delta t) &\approx v(x, y, t) - \Delta t(P_x + Q_y), \\ w(x, y, t + \Delta t) &\approx w(x, y, t) - \Delta t(H_x + R_y), \end{aligned} \tag{2.25}$$

with

$$\begin{aligned} F &= f + \frac{\Delta t}{2} f'(u)u_t + \frac{\Delta t^2}{6}\big(f''(u)(u_t)^2 + f'(u)u_{tt}\big), \\ G &= g + \frac{\Delta t}{2} g'(u)u_t + \frac{\Delta t^2}{6}\big(g''(u)(u_t)^2 + g'(u)u_{tt}\big), \end{aligned}$$

$$
\begin{aligned}
P &= f'(u)v + \frac{\Delta t}{2}\big(f''(u)u_t v + f'(u)v_t\big) + \frac{\Delta t^2}{6}\big(f'''(u)(u_t)^2 v + f''(u)(u_{tt}v + 2u_t v_t) + f'(u)v_{tt}\big),\\
Q &= g'(u)v + \frac{\Delta t}{2}\big(g''(u)u_t v + g'(u)v_t\big) + \frac{\Delta t^2}{6}\big(g'''(u)(u_t)^2 v + g''(u)(u_{tt}v + 2u_t v_t) + g'(u)v_{tt}\big),\\
H &= f'(u)w + \frac{\Delta t}{2}\big(f''(u)u_t w + f'(u)w_t\big) + \frac{\Delta t^2}{6}\big(f'''(u)(u_t)^2 w + f''(u)(u_{tt}w + 2u_t w_t) + f'(u)w_{tt}\big),\\
R &= g'(u)w + \frac{\Delta t}{2}\big(g''(u)u_t w + g'(u)w_t\big) + \frac{\Delta t^2}{6}\big(g'''(u)(u_t)^2 w + g''(u)(u_{tt}w + 2u_t w_t) + g'(u)w_{tt}\big).
\end{aligned}
$$

Now we apply the staggered central scheme strategy to (2.25) as in one dimension. Suppose at $t = t^n$, the approximations for the cell averages of the solution and the scaled cell averages of the first derivatives, denoted as $\{\bar{u}^n_{ij}\}_{ij}$, $\{\bar{v}^n_{ij}\}_{ij}$, and $\{\overline{w}^n_{ij}\}_{ij}$ are available on the primal mesh, that is, $\forall i, j$,

$$
\bar{u}^n_{ij} \approx \frac{1}{\Delta x \Delta y}\int_{I_{ij}} u\big(x, y, t^n\big)dxdy,
$$

$$
\bar{v}^n_{ij} \approx \frac{1}{\Delta y}\int_{I_{ij}} v\big(x, y, t^n\big)dxdy, \qquad \overline{w}^n_{ij} \approx \frac{1}{\Delta x}\int_{I_{ij}} w\big(x, y, t^n\big)dxdy.
$$

(b.1) We integrate (2.25) with $t = t^n$ over $I_{i+1/2,j+1/2}$, and approximate the cell averages of the solution and the scaled cell averages of the first derivatives with respect to the dual mesh at $t^{n+1} = t^n + \Delta t$, denoted as $\bar{u}^{n+1}_{i+1/2,j+1/2}$, $\bar{v}^{n+1}_{i+1/2,j+1/2}$ and $\overline{w}^{n+1}_{i+1/2,j+1/2}$, respectively, and this is given below.

$$
\begin{aligned}
\bar{u}^{n+1}_{i+1/2,j+1/2} = \bar{u}^n_{i+1/2,j+1/2} &- \frac{\Delta t}{\Delta x \Delta y}\left[\int_{y_j}^{y_{j+1}} \big(F\big(u\big(x_{i+1}, y, t^n\big)\big) - F\big(u\big(x_i, y, t^n\big)\big)\big)dy\right]\\
&- \frac{\Delta t}{\Delta x \Delta y}\left[\int_{x_i}^{x_{i+1}} \big(G\big(u\big(x, y_{j+1}, t^n\big)\big) - G\big(u\big(x, y_j, t^n\big)\big)\big)dx\right],
\end{aligned} \tag{2.26}
$$

$$
\begin{aligned}
\bar{v}^{n+1}_{i+1/2,j+1/2} = \bar{v}^n_{i+1/2,j+1/2}&\\
&- \frac{\Delta t}{\Delta y}\left[\int_{y_j}^{y_{j+1}} \big(P\big(u\big(x_{i+1}, y, t^n\big), v\big(x_{i+1}, y, t^n\big)\big) - P\big(u\big(x_i, y, t^n\big), v\big(x_i, y, t^n\big)\big)\big)dy\right]\\
&- \frac{\Delta t}{\Delta y}\left[\int_{x_i}^{x_{i+1}} \big(Q\,\big(u\big(x, y_{j+1}, t^n\big), v\big(x, y_{j+1}, t^n\big)\big) - Q\,\big(u\big(x, y_j, t^n\big), v\big(x, y_j, t^n\big)\big)\big)dx\right],
\end{aligned} \tag{2.27}
$$

$$
\begin{aligned}
\overline{w}^{n+1}_{i+1/2,j+1/2} = \overline{w}^n_{i+1/2,j+1/2}&\\
&- \frac{\Delta t}{\Delta x}\left[\int_{y_j}^{y_{j+1}} \big(H\big(u\big(x_{i+1}, y, t^n\big), w\big(x_{i+1}, y, t^n\big)\big) - H\big(u\big(x_i, y, t^n\big), w\big(x_i, y, t^n\big)\big)\big)dy\right]\\
&- \frac{\Delta t}{\Delta x}\left[\int_{x_i}^{x_{i+1}} \big(R\big(u\big(x, y_{j+1}, t^n\big), w\big(x, y_{j+1}, t^n\big)\big) - R\big(u\big(x, y_j, t^n\big), w\big(x, y_j, t^n\big)\big)\big)dx\right].
\end{aligned} \tag{2.28}
$$

(b.2) We then integrate (2.25) with $t = t^{n+1}$ over I_{ij}, and approximate the cell averages of the solution and the scaled cell averages of the first derivatives with respect to the primal mesh at $t^{n+2} = t^{n+1} + \Delta t$, denoted as $\bar{u}^{n+2}_{ij}$, $\bar{v}^{n+2}_{ij}$ and $\overline{w}^{n+2}_{ij}$, respectively. The formulations are omitted here due to the similarity to (2.26)–(2.28).

(b.3) Set n to be $n + 2$, and go to (b.1).

Note that the mesh switches back after two time steps.

From now on, we will focus on the update details from t^n to t^{n+1}. For the two-dimensional case, we will propose a fourth-order central HWENO method. With this in mind, we further approximate all the line integrals in (2.26)–(2.28) with the two-point Gaussian quadrature formula, and this will lead to our actual update formulations.

$$\begin{aligned}\bar{u}^{n+1}_{i+1/2,j+1/2} &= \bar{u}^{n}_{i+1/2,j+1/2} - \frac{\Delta t}{2\Delta x}\big[F^n_{i+1,j_l} + F^n_{i+1,j_r} - F^n_{i,j_l} - F^n_{i,j_r}\big] \\ &\quad - \frac{\Delta t}{2\Delta y}\big[G^n_{i_l,j+1} + G^n_{i_r,j+1} - G^n_{i_l,j} - G^n_{i_r,j}\big], \end{aligned} \tag{2.29}$$

$$\begin{aligned}\bar{v}^{n+1}_{i+1/2,j+1/2} &= \bar{v}^{n}_{i+1/2,j+1/2} - \frac{\Delta t}{2}\big[P^n_{i+1,j_l} + P^n_{i+1,j_r} - P^n_{i,j_l} - P^n_{i,j_r}\big] \\ &\quad - \frac{\Delta t\Delta x}{2\Delta y}\big[Q^n_{i_l,j+1} + Q^n_{i_r,j+1} - Q^n_{i_l,j} - Q^n_{i_r,j}\big], \end{aligned} \tag{2.30}$$

$$\begin{aligned}\bar{w}^{n+1}_{i+1/2,j+1/2} &= \bar{w}^{n}_{i+1/2,j+1/2} - \frac{\Delta t\Delta y}{2\Delta x}\big[H^n_{i+1,j_l} + H^n_{i+1,j_r} - H^n_{i,j_l} - H^n_{i,j_r}\big] \\ &\quad - \frac{\Delta t}{2}\big[R^n_{i_l,j+1} + R^n_{i_r,j+1} - R^n_{i_l,j} - R^n_{i_r,j}\big]. \end{aligned} \tag{2.31}$$

Here

$$\begin{aligned} i_l &= i + \frac{1}{2} - \frac{\sqrt{3}}{6}, \qquad i_r = i + \frac{1}{2} + \frac{\sqrt{3}}{6}, \\ j_l &= j + \frac{1}{2} - \frac{\sqrt{3}}{6}, \qquad j_r = j + \frac{1}{2} + \frac{\sqrt{3}}{6}, \end{aligned} \tag{2.32}$$

and the notation $p^n_{r,s}$ in (2.29)–(2.31) is for the point value $p(x_r, y_s, t^n)$ of a function p. When r or s is not an integer, x_r or y_s should be understood as a natural linear interpolation of $\{x_i\}_i$ or $\{y_j\}_j$ with respect to the subindex.

To obtain the cell averages $\bar{u}^{n+1}_{i+1/2,j+1/2}$, and the scaled cell averages $\bar{v}^{n+1}_{i+1/2,j+1/2}$ and $\bar{w}^{n+1}_{i+1/2,j+1/2}$ on the dual mesh at the next time t^{n+1} based on (2.29)–(2.31), one will need to reconstruct at the current time t^n, $\forall i, j$,

(1) the cell averages $\bar{u}^n_{i+1/2,j+1/2}$, and the scaled cell averages $\bar{v}^n_{i+1/2,j+1/2}$, $\bar{w}^n_{i+1/2,j+1/2}$ on the dual mesh, as well as
(2) the point value of $q(x_r, y_s, t^n)$, where $q = u, v, w, v_x, v_y, v_{xx}, v_{xy}, v_{yy}, w_x, w_y, w_{xx}, w_{xy}$, or w_{yy}, and they are the functions F, G, P, Q, H and R depend on. Here (x_r, y_s) is some *interior* point of the primal mesh, coming from the Gaussian quadrature nodes in each cell, and the collection of such points is denoted as

$$\mathcal{G} = \Big\{(x_i, y_{j-\frac{1}{2}+\frac{\sqrt{3}}{6}}), (x_i, y_{j+\frac{1}{2}-\frac{\sqrt{3}}{6}}), (x_{i-\frac{1}{2}+\frac{\sqrt{3}}{6}}, y_j), (x_{i+\frac{1}{2}-\frac{\sqrt{3}}{6}}, y_j), \ \forall i, j\Big\}. \tag{2.33}$$

Just as in one dimension, to obtain the cell average $\bar{u}^n_{i+1/2,j+1/2}$, $\forall i, j$, one would want to get the following four quarter cell averages

$$\begin{aligned} &\frac{1}{\Delta x\Delta y}\int\limits_{x_{i-1/2}}^{x_i}\int\limits_{y_{j-1/2}}^{y_j} u\big(x, y, t^n\big)dxdy, \quad \frac{1}{\Delta x\Delta y}\int\limits_{x_i}^{x_{i+1/2}}\int\limits_{y_{j-1/2}}^{y_j} u\big(x, y, t^n\big)dxdy, \\ &\frac{1}{\Delta x\Delta y}\int\limits_{x_{i-1/2}}^{x_i}\int\limits_{y_j}^{y_{j+1/2}} u\big(x, y, t^n\big)dxdy, \quad \frac{1}{\Delta x\Delta y}\int\limits_{x_i}^{x_{i+1/2}}\int\limits_{y_j}^{y_{j+1/2}} u\big(x, y, t^n\big)dxdy. \end{aligned} \tag{2.34}$$

This similarly goes to the scaled cell averages.

To reconstruct the quarter cell averages and point values mentioned above, we adapt the two-dimensional fourth-order accurate HWENO reconstruction of Zhu and Qiu [34], where only function values at some points located on the edges of each cell were reconstructed. Next in steps 1–3, the reconstruction will be described, and it is based on the (scaled) cell averages $\{\bar{u}^n_{ij}, \bar{v}^n_{ij}, \bar{w}^n_{ij}\}_{ij}$ on the primal mesh at $t = t^n$. The superscript n will be omitted, together with the dependence on the time t of u, v and w. To assist with the presentation, we relabel the cell I_{ij} and its neighboring cells as $I_1, \cdots, I_9$ as shown in Fig. 1. Note that I_{ij} is now I_5.

Step 1. A HWENO reconstruction of $\frac{1}{\Delta x\Delta y}\int_{x_{i-1/2}}^{x_i}\int_{y_{j-1/2}}^{y_j} u(x, y)dxdy$ from the (scaled) cell averages $\{\bar{u}_{ij}, \bar{v}_{ij}, \bar{w}_{ij}\}_{ij}$.

We first introduce eight “small” stencils

$$\begin{aligned} &S_1 = \{I_1, I_2, I_4, I_5\}, \qquad S_2 = \{I_2, I_3, I_5, I_6\}, \\ &S_3 = \{I_4, I_5, I_7, I_8\}, \qquad S_4 = \{I_5, I_6, I_8, I_9\}, \\ &S_5 = \{I_1, I_2, I_3, I_4, I_5, I_7\}, \qquad S_6 = \{I_1, I_2, I_3, I_5, I_6, I_9\}, \\ &S_7 = \{I_1, I_4, I_5, I_7, I_8, I_9\}, \qquad S_8 = \{I_3, I_5, I_6, I_7, I_8, I_9\}, \end{aligned}$$

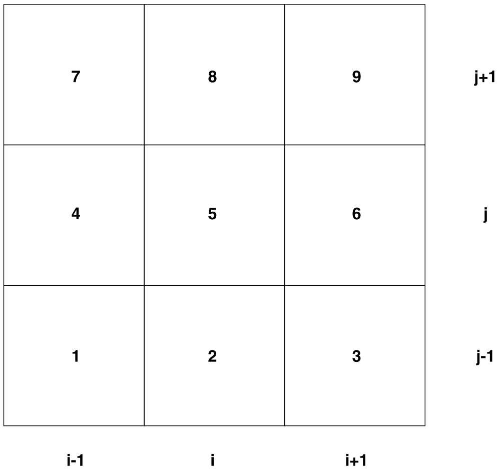

Fig. 1. New labels of the cell I_{ij} and its neighboring cells.

and reconstruct quadratic polynomials $p_1(x,y),\cdots,p_8(x,y)$ on $S_1,\cdots,S_8$, respectively. They all approximate the solution u, and satisfy

$$\frac{1}{\Delta x\Delta y}\int\limits_{I_k} p_n(x,y)dxdy=\bar{u}_k, \qquad \frac{1}{\Delta y}\int\limits_{I_{k_x}}\frac{\partial p_n(x,y)}{\partial x}dxdy=\bar{v}_{k_x},$$
$$\frac{1}{\Delta x}\int\limits_{I_{k_y}}\frac{\partial p_n(x,y)}{\partial y}dxdy=\overline{w}_{k_y}, \tag{2.35}$$

with

$$n=1,\quad k=1,2,4,5,\quad k_x=4,\quad k_y=2; \qquad n=2,\quad k=2,3,5,6,\quad k_x=6,\quad k_y=2;$$
$$n=3,\quad k=4,5,7,8,\quad k_x=4,\quad k_y=8; \qquad n=4,\quad k=5,6,8,9,\quad k_x=6,\quad k_y=8;$$

and

$$\frac{1}{\Delta x\Delta y}\int\limits_{I_k} p_n(x,y)dxdy=\bar{u}_k, \tag{2.36}$$

with

$$n=5,\quad k=1,2,3,4,5,7; \qquad n=6,\quad k=1,2,3,5,6,9;$$
$$n=7,\quad k=1,4,5,7,8,9; \qquad n=8,\quad k=3,5,6,7,8,9.$$

To combine the quadratic polynomials above to obtain a higher order approximation of fourth-order accuracy for a quarter cell average of u over $[x_{i-\frac{1}{2}},x_i]\times[y_{j-\frac{1}{2}},y_j]$, the linear weights $\gamma_1,\cdots,\gamma_8$ are chosen to ensure

$$\frac{1}{\Delta x\Delta y}\int\limits_{x_{i-1/2}}^{x_i}\int\limits_{y_{j-1/2}}^{y_j} u(x,y)dxdy=\frac{1}{\Delta x\Delta y}\sum_{n=1}^{8}\gamma_n\int\limits_{x_{i-1/2}}^{x_i}\int\limits_{y_{j-1/2}}^{y_j} p_n(x,y)dxdy \tag{2.37}$$

for any polynomial u of degree at most 3. Here, one should understand that $\bar{u}_k$, $\bar{v}_k$ and $\overline{w}_k$ on the right side of (2.35)–(2.36) are from the same polynomial u.

It is not hard to see that (2.37) holds for any polynomial u of degree at most 2 if $\sum_{n=1}^{8}\gamma_n=1$, as each individual $p_n(x)$ reconstructs a quadratic polynomial exactly. There are four more constraints on $\gamma_1,\cdots,\gamma_8$ from requiring (2.37) to hold for $u=x^3, x^2y, xy^2$ and y^3, respectively. Subject to these five constraints listed above, we determine the linear wights $\gamma_1,\cdots,\gamma_8$ by minimizing $\sum_{n=1}^{8}(\gamma_n)^2$. The obtained linear weights $\gamma_1,\cdots,\gamma_8$ turn out to be positive.

Similar to the one-dimensional case, smoothness indicators $\{\beta_n\}_n$ are needed to compute the nonlinear weights for the non-oscillatory HWENO reconstruction. Each smoothness indicator β_n measures how smooth the function $p_n(x)$ is in the target cell I_{ij}. The smaller the smoothness indicator β_n is, the smoother the function $p_n(x)$ is in I_{ij}. As in [8], the following smoothness indicator is taken,

$$\beta_n = \sum_{|l|=1}^{2} |I_{ij}|^{|l|-1} \int\limits_{I_{ij}} \left(\frac{\partial^{|l|}}{\partial x^{l_1} \partial y^{l_2}} p_n(x,y) \right)^2 dxdy,$$

where $l = (l_1, l_2)$ and $|l| = l_1 + l_2$. With the smoothness indicators and the linear weights, we can get the nonlinear weights $\omega_n, n = 1, \cdots, 8$,

$$\omega_n = \frac{\overline{\omega}_n}{\sum_{k=1}^{8} \overline{\omega}_k}, \quad \text{where } \overline{\omega}_k = \frac{\gamma_k}{(\varepsilon + \beta_k)^2},\ k = 1, \cdots, 8.$$

Finally, a fourth-order HWENO approximation is reconstructed for a quarter cell average of u,

$$\frac{1}{\Delta x \Delta y} \int\limits_{x_{i-1/2}}^{x_i} \int\limits_{y_{j-1/2}}^{y_j} u(x,y)dxdy \approx \frac{1}{\Delta x \Delta y} \sum_{n=1}^{8} \omega_n \int\limits_{x_{i-1/2}}^{x_i} \int\limits_{y_{j-1/2}}^{y_j} p_n(x,y)dxdy.$$

Following the similar procedure, one can reconstruct the other quarter cell averages in (2.34). In actual implementation, one only needs to reconstruct three quarter cell averages in the target cell I_{ij}, and the fourth one can be computed based on the local conservation. For example, once the first three in (2.34) are available, the fourth one can be obtained as

$$\frac{1}{\Delta x \Delta y} \int\limits_{x_i}^{x_{i+1/2}} \int\limits_{y_j}^{y_{j+1/2}} u(x,y)dxdy \approx \bar{u}_{ij} - \frac{1}{\Delta x \Delta y} \left(\int\limits_{x_{i-1/2}}^{x_i} \int\limits_{y_{j-1/2}}^{y_j} u(x,y)dxdy \right.$$
$$\left. + \int\limits_{x_i}^{x_{i+1/2}} \int\limits_{y_{j-1/2}}^{y_j} u(x,y)dxdy + \int\limits_{x_{i-1/2}}^{x_i} \int\limits_{y_j}^{y_{j+1/2}} u(x,y)dxdy \right).$$

The reconstructions of the point values $u(x_*, y_*)$ with $(x_*, y_*) \in \mathcal{G}$ can be proceeded similarly, and the details are omitted here. One can refer to [26] for the related details of reconstructing the point values of u on edges of a cell.

Step 2. A HWENO reconstruction of $\frac{1}{\Delta y} \int_{x_{i-1/2}}^{x_i} \int_{y_{j-1/2}}^{y_j} v(x,y)dxdy$ from the (scaled) cell averages $\{\bar{u}_{ij}, \bar{v}_{ij}, \overline{w}_{ij}\}_{ij}$.

In this step, the same stencils $S_1, \cdots, S_8$ as in step 1 of this section are used. We first construct four cubic polynomials $p_n(x,y)$ on S_n, $n = 1, \cdots, 4$, which approximate the solution u and are determined by the following,

$$\frac{1}{\Delta x \Delta y} \int\limits_{I_k} p_n(x,y)dxdy = \bar{u}_k, \qquad \frac{1}{\Delta y} \int\limits_{I_{k_x}} \frac{\partial p_n(x,y)}{\partial x} dxdy = \bar{v}_{k_x},$$
$$\frac{1}{\Delta x} \int\limits_{I_{k_y}} \frac{\partial p_n(x,y)}{\partial y} dxdy = \overline{w}_{k_y},$$

with

$$n = 1, \quad k = 1,2,4,5, \quad k_x = 1,4,5, \quad k_y = 1,2,5; \qquad n = 2, \quad k = 2,3,5,6, \quad k_x = 3,5,6, \quad k_y = 2,3,5;$$
$$n = 3, \quad k = 4,5,7,8, \quad k_x = 4,5,7, \quad k_y = 5,7,8; \qquad n = 4, \quad k = 5,6,8,9, \quad k_x = 5,6,9, \quad k_y = 5,8,9.$$

We then construct four more quadratic polynomials $p_n(x,y)$ on S_n, $n = 5, \cdots, 8$, which approximate $v = u_x$ and satisfy

$$\frac{1}{\Delta y} \int\limits_{I_{k_x}} p_n(x,y)dxdy = \bar{v}_{k_x},$$

with

$$n = 5, \quad k_x = 1,2,3,4,5,7; \qquad n = 6, \quad k_x = 1,2,3,5,6,9;$$
$$n = 7, \quad k_x = 1,4,5,7,8,9; \qquad n = 8, \quad k_x = 3,5,6,7,8,9.$$

In order to obtain a fourth-order approximation for the solution u, the approximations to the derivative of u only need to be third-order accurate. From the definitions of $\{p_n(x,y)\}_n$, one can easily see that each of the following

$$\frac{1}{\Delta y} \int\limits_{x_{i-1/2}}^{x_i} \int\limits_{y_{j-1/2}}^{y_j} \frac{\partial p_n(x,y)}{\partial x} dxdy, \quad n = 1, \cdots, 4, \qquad \frac{1}{\Delta y} \int\limits_{x_{i-1/2}}^{x_i} \int\limits_{y_{j-1/2}}^{y_j} p_n(x,y)dxdy, \quad n = 5, \cdots, 8$$

provides a fourth-order approximation of $\frac{1}{\Delta y}\int_{x_{i-1/2}}^{x_i}\int_{y_{j-1/2}}^{y_j} v(x,y)dxdy$ for the smooth function u (here $v = u_x$), so does any convex combination of these eight quantities. In this paper, we choose the simple linear weights $\gamma_{xn} = \frac{1}{8}$, $n = 1, \cdots, 8$. As for the smoothness indicators, we follow [26,34] and take

$$\beta_n = \sum_{|l|=1}^{2} |I_{ij}|^{|l|} \int_{I_{ij}} \left(\frac{\partial^{|l|}}{\partial x^{l_1} \partial y^{l_2}} \frac{\partial}{\partial x} p_n(x,y) \right)^2 dxdy, \quad n = 1, \cdots, 4,$$

$$\beta_n = \sum_{|l|=1}^{2} |I_{ij}|^{|l|} \int_{I_{ij}} \left(\frac{\partial^{|l|}}{\partial x^{l_1} \partial y^{l_2}} p_n(x,y) \right)^2 dxdy, \quad n = 5, \cdots, 8,$$

where $l = (l_1, l_2)$ and $|l| = l_1 + l_2$. With the smoothness indicators and the linear weights, we can compute the nonlinear weights ω_n, $n = 1, \cdots, 8$,

$$\omega_n = \frac{\overline{\omega}_n}{\sum_{k=1}^{8} \overline{\omega}_k}, \quad \text{where } \overline{\omega}_k = \frac{\gamma_{xk}}{(\varepsilon + \beta_k)^2}, \; k = 1, \cdots, 8.$$

Now we are ready to reconstruct a fourth-order HWENO approximation for a scaled quarter cell average of v,

$$\frac{1}{\Delta y} \int_{x_{i-1/2}}^{x_i} \int_{y_{j-1/2}}^{y_j} v(x,y)dxdy \approx \frac{1}{\Delta y} \int_{x_{i-1/2}}^{x_i} \int_{y_{j-1/2}}^{y_j} \left[\sum_{n=1}^{4} \omega_n \frac{\partial p_n(x,y)}{\partial x} + \sum_{n=5}^{8} \omega_n p_n(x,y) \right] dxdy.$$

The reconstructions of the other scaled quarter cell averages are similar. Just as in step 1, in practice, one only needs to reconstruct any three scaled quarter cell averages of v in the target cell I_{ij}, and the fourth one can be obtained based on the local conservation.

The reconstructions of the point values $q(x_*, y_*)$ with $q = v, v_x, v_y, v_{xx}, v_{xy}, v_{yy}$ and $(x_*, y_*) \in \mathcal{G}$ can be proceeded similarly, and one can refer to [34] for the related details of reconstructing the point values of v on edges of a cell. We here only point out the differences. To approximate the point values of v_x and v_y, the difference is in the smoothness indicators. In particular, for $v_x(x_*, y_*)$, we take

$$\beta_n = \sum_{|l|=1} |I_{ij}|^2 \int_{I_{ij}} \left(\frac{\partial}{\partial x^{l_1} \partial y^{l_2}} \frac{\partial^2}{\partial x^2} p_n(x,y) \right)^2 dxdy, \quad n = 1, \cdots, 4,$$

$$\beta_n = \sum_{|l|=1} |I_{ij}|^2 \int_{I_{ij}} \left(\frac{\partial}{\partial x^{l_1} \partial y^{l_2}} \frac{\partial}{\partial x} p_n(x,y) \right)^2 dxdy, \quad n = 5, \cdots, 8,$$

and for $v_y(x_*, y_*)$, we take

$$\beta_n = \sum_{|l|=1} |I_{ij}|^2 \int_{I_{ij}} \left(\frac{\partial}{\partial x^{l_1} \partial y^{l_2}} \frac{\partial^2}{\partial x \partial y} p_n(x,y) \right)^2 dxdy, \quad n = 1, \cdots, 4,$$

$$\beta_n = \sum_{|l|=1} |I_{ij}|^2 \int_{I_{ij}} \left(\frac{\partial}{\partial x^{l_1} \partial y^{l_2}} \frac{\partial}{\partial y} p_n(x,y) \right)^2 dxdy, \quad n = 5, \cdots, 8,$$

where $l = (l_1, l_2)$ and $|l| = l_1 + l_2$. To approximate v_{xx}, v_{xy}, v_{yy} at (x_*, y_*), the linear weights rather than the nonlinear weights are used in the final reconstructions.

Step 3. A HWENO reconstruction of $\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i}\int_{y_{j-1/2}}^{y_j} w(x,y)dxdy$ from the (scaled) cell averages $\{\bar{u}_{ij}, \bar{v}_{ij}, \overline{w}_{ij}\}_{ij}$.

In this step, the same stencils $S_1, \cdots, S_8$ as in step 1 of this section are used. We first construct four cubic polynomials $p_n(x,y)$ on S_n, $n = 1, \cdots, 4$, which approximate the solution u and are determined by the following

$$\frac{1}{\Delta x \Delta y} \int_{I_k} p_n(x,y)dxdy = \bar{u}_k, \qquad \frac{1}{\Delta y} \int_{I_{k_x}} \frac{\partial p_n(x,y)}{\partial x} dxdy = \bar{v}_{k_x},$$

$$\frac{1}{\Delta x} \int_{I_{k_y}} \frac{\partial p_n(x,y)}{\partial y} dxdy = \overline{w}_{k_y},$$

with

$$n=1,\quad k=1,2,4,5,\quad k_x=1,4,5,\quad k_y=1,2,5;\qquad n=2,\quad k=2,3,5,6,\quad k_x=3,5,6,\quad k_y=2,3,5;$$

$$n=3,\quad k=4,5,7,8,\quad k_x=4,5,7,\quad k_y=5,7,8;\qquad n=4,\quad k=5,6,8,9,\quad k_x=5,6,9,\quad k_y=5,8,9.$$

We then construct four more quadratic polynomials $p_n(x,y)$ on S_n, $n=5,\cdots,8$, which approximate $w=u_y$ and satisfy

$$\frac{1}{\Delta x}\int\limits_{I_{k_y}} p_n(x,y)dxdy=\overline{w}_{k_y},$$

with

$$n=5,\quad k_y=1,2,3,4,5,7;\qquad n=6,\quad k_y=1,2,3,5,6,9;$$

$$n=7,\quad k_y=1,4,5,7,8,9;\qquad n=8,\quad k_y=3,5,6,7,8,9.$$

In order to obtain a fourth-order approximation for the solution u, the approximations to the derivative of u only need to be third-order accurate. From the definitions of $\{p_n(x,y)\}_n$, one can easily see that each of the following

$$\frac{1}{\Delta x}\int\limits_{x_{i-1/2}}^{x_i}\int\limits_{y_{j-1/2}}^{y_j}\frac{\partial p_n(x,y)}{\partial y}dxdy,\quad n=1,\cdots,4,;\qquad \frac{1}{\Delta x}\int\limits_{x_{i-1/2}}^{x_i}\int\limits_{y_{j-1/2}}^{y_j} p_n(x,y)dxdy,\quad n=5,\cdots,8$$

provides a fourth-order approximation of $\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_i}\int_{y_{j-1/2}}^{y_j} w(x,y)dxdy$ for the smooth function u, again here $w=u_y$. In this paper, we choose the simple linear weights $\gamma_{yn}=\frac{1}{8}$, $n=1,\cdots,8$. As for the smoothness indicators, we take

$$\beta_n=\sum_{|l|=1}^{2}|I_{ij}|^{|l|}\int\limits_{I_{ij}}\left(\frac{\partial^{|l|}}{\partial x^{l_1}\partial y^{l_2}}\frac{\partial}{\partial y}p_n(x,y)\right)^2dxdy,\quad n=1,\cdots,4,$$

$$\beta_n=\sum_{|l|=1}^{2}|I_{ij}|^{|l|}\int\limits_{I_{ij}}\left(\frac{\partial^{|l|}}{\partial x^{l_1}\partial y^{l_2}}p_n(x,y)\right)^2dxdy,\quad n=5,\cdots,8,$$

where $l=(l_1,l_2)$ and $|l|=l_1+l_2$. With the smoothness indicators and the linear weights, we can compute the nonlinear weights ω_n, $n=1,\cdots,8$,

$$\omega_n=\frac{\overline{\omega}_n}{\sum_{k=1}^{8}\overline{\omega}_k},\quad \text{where } \overline{\omega}_k=\frac{\gamma_{yk}}{(\varepsilon+\beta_k)^2},\ k=1,\cdots,8.$$

Now we are ready to reconstruct a fourth-order HWENO approximation for a scaled quarter cell average of w,

$$\frac{1}{\Delta x}\int\limits_{x_{i-1/2}}^{x_i}\int\limits_{y_{j-1/2}}^{y_j} w(x,y)dxdy\approx\frac{1}{\Delta x}\int\limits_{x_{i-1/2}}^{x_i}\int\limits_{y_{j-1/2}}^{y_j}\left[\sum_{n=1}^{4}\omega_n\frac{\partial p_n(x,y)}{\partial y}+\sum_{n=5}^{8}\omega_n p_n(x,y)\right]dxdy.$$

The reconstructions of the other scaled quarter cell averages are similar. Again, one of the scaled quarter cell average in the target cell I_{ij} can be computed from the others based on the local conservation.

The reconstructions of the point values $q(x_*,y_*)$ with $q=w,w_x,w_y,w_{xx},w_{xy},w_{yy}$ and $(x_*,y_*)\in\mathcal{G}$ can be proceeded similarly, and one can refer to [34] for the related details of reconstructing the point values of w on edges of a cell. We here only point out the differences. To approximate the point values of w_x and w_y, the difference is in the smoothness indicators. In particular, for $w_x(x_*,y_*)$, we take

$$\beta_n=\sum_{|l|=1}|I_{ij}|^2\int\limits_{I_{ij}}\left(\frac{\partial}{\partial x^{l_1}\partial y^{l_2}}\frac{\partial^2}{\partial x\partial y}p_n(x,y)\right)^2dxdy,\quad n=1,\cdots,4,$$

$$\beta_n=\sum_{|l|=1}|I_{ij}|^2\int\limits_{I_{ij}}\left(\frac{\partial}{\partial x^{l_1}\partial y^{l_2}}\frac{\partial}{\partial x}p_n(x,y)\right)^2dxdy,\quad n=5,\cdots,8,$$

and for $w_y(x_*,y_*)$, we take

$$\beta_n=\sum_{|l|=1}|I_{ij}|^2\int\limits_{I_{ij}}\left(\frac{\partial}{\partial x^{l_1}\partial y^{l_2}}\frac{\partial^2}{\partial y^2}p_n(x,y)\right)^2dxdy,\quad n=1,\cdots,4,$$

$$\beta_n=\sum_{|l|=1}|I_{ij}|^2\int\limits_{I_{ij}}\left(\frac{\partial}{\partial x^{l_1}\partial y^{l_2}}\frac{\partial}{\partial y}p_n(x,y)\right)^2dxdy,\quad n=5,\cdots,8,$$

where $l = (l_1, l_2)$ and $|l| = l_1 + l_2$. To approximate w_{xx}, w_{xy}, w_{yy} at (x_*, y_*), the linear weights rather than the nonlinear weights are used in the final reconstructions.

Remark 2. Similar as in the one-dimensional case (see also Remark 1), with the current central scheme framework, there is no need to use numerical fluxes for F, G, P, Q, H and R as in upwind type schemes.

Finally in this section, we briefly discuss the system case. Just as in one dimension, to reduce the spurious oscillations and enhance the numerical stability, we apply the local characteristic decomposition during the reconstructions of the quarter cell averages for u, v, w based on $\{\bar{u}_{ij}, \bar{v}_{ij}, \overline{w}_{ij}\}_{ij}$. The reconstructions of the point values are still implemented through a componentwise procedure. Note that for the system case in two dimensions, there are two flux functions $f(u)$ and $g(u)$. In the reconstructions of quarter cell averages, it is not obvious to know the direction along which the information is propagating locally. In our simulation, we adopt the strategy that we first perform the reconstruction based on the local characteristic information provided by $f(u)$ and $g(u)$, respectively. The final approximation is then given by an arithmetic average of these two reconstructions. Again to assist with the derivation of the proposed methods, MATLAB Symbolic Math Toolbox is used.

3. Central Hermite WENO scheme with natural continuous extension of Runge–Kutta time discretization

Alternative to the Lax–Wendroff strategy, we apply in this section the natural continuous extension of Runge–Kutta (NCE-RK) time discretizations to combine with the central HWENO spatial discretizations in the framework of staggered meshes. The use of NCE-RK methods permits one to compute accurate approximations for the intermediate value of a solution to an ODE based on standard RK methods with slight increase of the computational time. Note that a standard RK method only provides accurate approximations for the solution at discrete time t^n for any $\forall n$.

Below we will describe a fourth-order NCE-RK method which is used in this paper. For more details about such time discretizations, one can refer to [33,3]. Consider an ODE problem

$$\begin{cases} y'(t) = F\big(t, y(t)\big), \\ y(t_0) = y_0, \end{cases} \tag{3.1}$$

and suppose y^n is a given approximation to $y(t^n)$. One can then approximate $y(t^{n+1})$ at $t^{n+1} = t^n + \Delta t$ by y^{n+1} with a standard four-stage fourth-order RK scheme as follows.

$$y^{n+1} = y^n + \Delta t \sum_{i=1}^{4} b_i K^{(i)}, \tag{3.2}$$

where $K^{(i)}$ is an RK flux determined by

$$K^{(i)} = F\big(t^n + c_i \Delta t, Y^{(i)}\big) \quad \text{with } Y^{(i)} = y^n + c_i \Delta t K^{(i-1)},\ i = 1, 2, 3, 4, \tag{3.3}$$

and $K^{(0)} = 0$. In addition, $b_1 = b_4 = \frac{1}{6}$, $b_2 = b_3 = \frac{1}{3}$, and $c_1 = 0$, $c_2 = c_3 = \frac{1}{2}$, $c_4 = 1$.

A natural continuous extension of the RK scheme (3.2)–(3.3) further provides an approximation of $y(t)$ (and also its derivatives) with the same accuracy when $t \in [t^n, t^{n+1}]$. This approximation is given specifically by

$$s(t)|_{t=t^n+\theta\Delta t} := y^n + \Delta t \sum_{i=1}^{4} B_i(\theta) K^{(i)}, \quad 0 \le \theta \le 1,$$

where

$$B_1(\theta) = 2(1 - 4b_1)\theta^3 + 3(3b_1 - 1)\theta^2 + \theta,$$
$$B_i(\theta) = 4(3c_i - 2)b_i\theta^3 + 3(3 - 4c_i)b_i\theta^2, \quad i = 2, 3, 4.$$

$s(t)$ not only satisfies $s(t^n) = y^n$, $s(t^{n+1}) = y^{n+1}$, but also has the following approximation properties,

$$\max_{t^n \le t \le t^n+\Delta t} \big|y^{(l)}(t) - s^{(l)}(t)\big| = O\big(\Delta t^{4-l}\big), \quad 0 \le l \le 4.$$

3.1. One-dimensional case

We use the same notation for the staggered meshes and the cell averages of functions, as well as the fifth-order HWENO spatial reconstruction as in Section 2.1. Though the discussion below focuses on one time step, one would want to keep in mind that the overall algorithm is still based on staggered meshes, and it switches back and forth between the primal and dual meshes.

Suppose at $t = t^n$, the approximations for the cell averages of the solution and its first derivative, namely $\{\bar{u}_i^n\}_i$ and $\{\bar{v}_i^n\}_i$, are available on the primal mesh. We integrate (2.1)–(2.2) over $[t^n, t^{n+1}] \times [x_i, x_{i+1}]$ and get

$$\bar{u}_{i+1/2}^{n+1} = \bar{u}_{i+1/2}^{n} - \frac{1}{\Delta x}\int_{t^n}^{t^{n+1}} \left[f\big(u(x_{i+1},t)\big) - f\big(u(x_i,t)\big)\right]dt,$$
$$\bar{v}_{i+1/2}^{n+1} = \bar{v}_{i+1/2}^{n} - \frac{1}{\Delta x}\int_{t^n}^{t^{n+1}} \left[g\big(u(x_{i+1},t), v(x_{i+1},t)\big) - g\big(u(x_i,t), v(x_i,t)\big)\right]dt, \tag{3.4}$$

where $g(u,v) = f'(u)v$. If we consider the problem (2.1)–(2.2) with the piecewise initial data $(u,v)|_{I_i} = (\bar{u}_i^n, \bar{v}_i^n)$ at t^n, under the assumption that the time step Δt satisfies a CFL restriction $\Delta t \le \frac{1}{2}\frac{\Delta x}{\max|f'(u)|}$, the discontinuities starting at t^n from $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$ will not propagate to x_i over a single time step Δt, and therefore the solutions of this problem restricted at x_i, $\forall i$ are smooth for $t \in (t^n, t^{n+1})$. Motivated by this, the temporal integration in (3.4) can be further approximated using numerical quadratures with compatible accuracy. To match the fifth-order HWENO reconstruction to be used in the spatial discretization, we apply the three-point Gaussian quadrature formula and replace the temporal integral terms in (3.4) according to the following

$$\int_{t^n}^{t^{n+1}} f\big(u(x_i,t)\big)dt \approx \Delta t \sum_{l=1}^{3} \alpha_l f\big(u(x_i, t^n + \Delta t\theta_l)\big),$$
$$\int_{t^n}^{t^{n+1}} g\big(u(x_i,t), v(x_i,t)\big)dt \approx \Delta t \sum_{l=1}^{3} \alpha_l g\big(u(x_i, t^n + \Delta t\theta_l), v(x_i, t^n + \Delta t\theta_l)\big). \tag{3.5}$$

Here $\alpha_1 = \alpha_3 = \frac{5}{18}, \alpha_2 = \frac{4}{9}$, and $\theta_1 = \frac{1}{2} - \frac{\sqrt{15}}{10}, \theta_2 = \frac{1}{2}, \theta_3 = \frac{1}{2} + \frac{\sqrt{15}}{10}$ are the weights and quadrature points.

Based on (3.4) and (3.5), one can compute the cell averages of the solution and its derivative, namely $\{\bar{u}_{i+1/2}^{n+1}\}_i$ and $\{\bar{v}_{i+1/2}^{n+1}\}_i$, with respect to the dual mesh at $t = t^{n+1}$. To achieve this, one only needs to obtain accurate approximations for

$$\bar{u}_{i+1/2}^n, \bar{v}_{i+1/2}^n, u\big(x_i, t^n + \Delta t\theta_l\big), v\big(x_i, t^n + \Delta t\theta_l\big), \quad l = 1,2,3, \ \forall i, \tag{3.6}$$

and the remainder of this subsection will be devoted to the related details.

Recall at t^n, $\{\bar{u}_i^n, \bar{v}_i^n\}_i$ are available. We can first reconstruct the staggered cell averages $\{\bar{u}_{i+1/2}^n\}_i$ and $\{\bar{v}_{i+1/2}^n\}_i$ in (3.6) based on the same fifth-order HWENO procedure described in Section 2.1. In order to approximate the point values $u(x_i, t^n + \Delta t\theta_l)$, $v(x_i, t^n + \Delta t\theta_l)$, $l = 1,2,3$, $\forall i$ in (3.6), we will apply the fourth-order NCE-RK method to the auxiliary ODE problem (3.1) starting from $t = t^n$ at each x_i with any i, where

$$y(t) = \begin{cases} u(x_i,t) \\ v(x_i,t), \end{cases} \qquad F(t,y) = \begin{cases} -f'(u)v|_{x_i} \\ -f''(u)v^2 - f'(u)v_x|_{x_i} \approx -f''(u)v^2 - f'(u)\mathcal{R}_{v_x}|_{x_i}. \end{cases} \tag{3.7}$$

More specifically, one needs to evaluate the corresponding $K^{(i)}$, $i = 1,2,3,4$ used in (3.2)–(3.3). Here the operator $\mathcal{R}_{v_x}$ in (3.7) is to reconstruct v_x at x_i, and it will be specified.

To evaluate $K^{(1)}$, we reconstruct the point value of $y(t^n)$, namely $u(x_i,t^n)$ and $v(x_i,t^n)$, based on the given cell averages $\{\bar{u}_i^n, \bar{v}_i^n\}_i$ on the primal mesh. The operator $\mathcal{R}_{v_x}$ is chosen such that $v_x(x_i,t^n)$ is also reconstructed from the same set of cell averages. These reconstructions have already been discussed in Section 2.1 as steps 2, 4, 5. Once $K^{(l)}$ for some $l \ge 1$ is available, then $Y^{(l+1)}$ can be computed based on (3.3). This will provide approximations for $u(x_i, t^n + c_{l+1}\Delta t)$ and $v(x_i, t^n + c_{l+1}\Delta t)$ for all i. Now the operator $\mathcal{R}_{v_x}$ in (3.7) is chosen such that $v_x(x_i, t^n + c_{l+1}\Delta t)$ is reconstructed based on these approximated point values $\{u(x_i, t^n + c_{l+1}\Delta t), v(x_i, t^n + c_{l+1}\Delta t)\}_i$ using a HWENO strategy similar to the step 5 in Section 2.1, see also [17,23] for more details. With $K^{(l)}$, $l = 1,2,3,4$, one can now follow the NCE-RK procedure to obtain accurate approximations for the point values of u and v in (3.6).

3.2. Two-dimensional case

We use the same notation for the staggered meshes and the cell averages of functions, as well as the fourth-order HWENO spatial reconstruction as in Section 2.2. Again, the discussion will focus on the algorithm over one time step.

Suppose at $t = t^n$, the approximations for the cell averages of the solution and its first derivative, namely $\{\bar{u}^n_{ij}, \bar{v}^n_{ij}, \overline{w}^n_{ij}\}_{ij}$, are available on the primal mesh. We integrate (2.21)–(2.23) over $[t^n, t^{n+1}] \times [x_i, x_{i+1}] \times [y_j, y_{j+1}]$ and get

$$\begin{aligned}
\bar{u}^{n+1}_{i+1/2,j+1/2} &= \bar{u}^n_{i+1/2,j+1/2} \\
&\quad - \frac{1}{\Delta x \Delta y}\left[\int_{t^n}^{t^{n+1}}\int_{y_j}^{y_{j+1}} \big(f\big(u(x_{i+1},y,t)\big) - f\big(u(x_i,y,t)\big)\big)dydt\right] \\
&\quad - \frac{1}{\Delta x \Delta y}\left[\int_{t^n}^{t^{n+1}}\int_{x_i}^{x_{i+1}} \big(g\big(u(x,y_{j+1},t)\big) - g\big(u(x,y_j,t)\big)\big)dxdt\right],
\end{aligned} \tag{3.8}$$

$$\begin{aligned}
\bar{v}^{n+1}_{i+1/2,j+1/2} &= \bar{v}^n_{i+1/2,j+1/2} \\
&\quad - \frac{1}{\Delta y}\left[\int_{t^n}^{t^{n+1}}\int_{y_j}^{y_{j+1}} \big(p\big(u(x_{i+1},y,t), v(x_{i+1},y,t)\big) - p\big(u(x_i,y,t), v(x_i,y,t)\big)\big)dydt\right] \\
&\quad - \frac{1}{\Delta y}\left[\int_{t^n}^{t^{n+1}}\int_{x_i}^{x_{i+1}} \big(q\big(u(x,y_{j+1},t), v(x,y_{j+1},t)\big) - q\big(u(x,y_j,t), v(x,y_j,t)\big)\big)dxdt\right],
\end{aligned} \tag{3.9}$$

$$\begin{aligned}
\overline{w}^{n+1}_{i+1/2,j+1/2} &= \overline{w}^n_{i+1/2,j+1/2} \\
&\quad - \frac{1}{\Delta x}\left[\int_{t^n}^{t^{n+1}}\int_{y_j}^{y_{j+1}} \big(h\big(u(x_{i+1},y,t), w(x_{i+1},y,t)\big) - h\big(u(x_i,y,t), w(x_i,y,t)\big)\big)dydt\right] \\
&\quad - \frac{1}{\Delta x}\left[\int_{t^n}^{t^{n+1}}\int_{x_i}^{x_{i+1}} \big(r\big(u(x,y_{j+1},t), w(x,y_{j+1},t)\big) - r\big(u(x,y_j,t), w(x,y_j,t)\big)\big)dxdt\right],
\end{aligned} \tag{3.10}$$

where

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

Since a fourth-order HWENO spatial reconstruction will be used, we further apply the two-point Gaussian quadrature formula to approximate the spatial integration in (3.8)–(3.10) and obtain

$$\begin{aligned}
\bar{u}^{n+1}_{i+1/2,j+1/2} &\approx \bar{u}^n_{i+1/2,j+1/2} - \frac{1}{2\Delta x}\int_{t^n}^{t^{n+1}} \big(f_{i+1,j_l}(t) + f_{i+1,j_r}(t) - f_{i,j_l}(t) - f_{i,j_r}(t)\big)dt \\
&\quad - \frac{1}{2\Delta y}\int_{t^n}^{t^{n+1}} \big(g_{i_l,j+1}(t) + g_{i_r,j+1}(t) - g_{i_l,j}(t) - g_{i_r,j}(t)\big)dt,
\end{aligned} \tag{3.11}$$

$$\begin{aligned}
\bar{v}^{n+1}_{i+1/2,j+1/2} &\approx \bar{v}^n_{i+1/2,j+1/2} - \frac{1}{2}\int_{t^n}^{t^{n+1}} \big(p_{i+1,j_l}(t) + p_{i+1,j_r}(t) - p_{i,j_l}(t) - p_{i,j_r}(t)\big)dt \\
&\quad - \frac{\Delta x}{2\Delta y}\int_{t^n}^{t^{n+1}} \big(q_{i_l,j+1}(t) + q_{i_r,j+1}(t) - q_{i_l,j}(t) - q_{i_r,j}(t)\big)dt,
\end{aligned} \tag{3.12}$$

$$\begin{aligned}
\overline{w}^{n+1}_{i+1/2,j+1/2} &\approx \overline{w}^n_{i+1/2,j+1/2} - \frac{\Delta y}{2\Delta x}\int_{t^n}^{t^{n+1}} \big(h_{i+1,j_l}(t) + h_{i+1,j_r}(t) - h_{i,j_l}(t) - h_{i,j_r}(t)\big)dt \\
&\quad - \frac{1}{2}\int_{t^n}^{t^{n+1}} \big(r_{i_l,j+1}(t) + r_{i_r,j+1}(t) - r_{i_l,j}(t) - r_{i_r,j}(t)\big)dt.
\end{aligned} \tag{3.13}$$

Here the notation $p_{r,s}(t)$ in (3.11)–(3.13) is for the point value of $p(x_r, y_s, t)$ of a function p, with i_l, i_r, j_l, j_r given in (2.32). The collection of the related points is denoted as $\mathcal{G}$

$$\mathcal{G} = \Big\{(x_i, y_j), (x_i, y_{j-\frac{1}{2}+\frac{\sqrt{3}}{6}}), (x_i, y_{j+\frac{1}{2}-\frac{\sqrt{3}}{6}}), (x_{i-\frac{1}{2}+\frac{\sqrt{3}}{6}}, y_j), (x_{i+\frac{1}{2}-\frac{\sqrt{3}}{6}}, y_j), \ \forall i, j\Big\}. \tag{3.14}$$

If the time step Δt is chosen to satisfy the CFL restriction $\Delta t \le \frac{C_{cfl}}{(\max|f'(u)|/\Delta x + \max|g'(u)|/\Delta y)}$ with some constant C_{cfl}, one can expect $p(x_*, y_*, t)$, with $(x_*, y_*) \in \mathcal{G}$ and $p = u, v, w$, are smooth with respect to $t \in (t^n, t^{n+1})$. Under such assumption on the time step, we further apply the three-point Gaussian quadrature formula and replace the temporal integral terms in (3.11)–(3.13) according to the following

$$\int_{t^n}^{t^{n+1}} s(\cdot, t)dt \approx \Delta t \sum_{l=1}^{3} \alpha_l s\big(\cdot, t^n + \Delta t \theta_l\big), \qquad s = f, g, p, q, h, r, \tag{3.15}$$

where $\alpha_1 = \alpha_3 = \frac{5}{18}, \alpha_2 = \frac{4}{9}$, and $\theta_1 = \frac{1}{2} - \frac{\sqrt{15}}{10}, \theta_2 = \frac{1}{2}, \theta_3 = \frac{1}{2} + \frac{\sqrt{15}}{10}$.

Based on (3.8)–(3.15), one can compute the cell averages of the solution and its derivative, namely $\{\bar{u}^{n+1}_{i+1/2,j+1/2}, \bar{v}^{n+1}_{i+1/2,j+1/2}, \bar{w}^{n+1}_{i+1/2,j+1/2}\}_{ij}$, with respect to the dual mesh at $t = t^{n+1}$. To achieve this, one only needs to obtain accurate approximations for

$$\bar{p}^n_{i+1/2,j+1/2}, \, p\big(x_*, y_*, t^n + \Delta t \theta_l\big), \quad \text{with } p = u, v, w \text{ and } (x_*, y_*) \in \mathcal{G}, \ l = 1, 2, 3, \ \forall i, j \tag{3.16}$$

and the remainder of this subsection will be devoted to the related details.

Recall at t^n, $\{\bar{u}^n_{ij}, \bar{v}^n_{ij}, \bar{w}^n_{ij}\}_{ij}$ are available. We can first reconstruct the staggered cell averages $\{\bar{u}^n_{i+1/2,j+1/2}, \bar{v}^n_{i+1/2,j+1/2}, \bar{w}^n_{i+1/2,j+1/2}\}_{ij}$ in (3.16) based on the same fourth-order HWENO procedure described in Section 2.2. In order to approximate the point values $p(x_*, y_*, t^n + \Delta t \theta_l)$ with $p = u, v, w$ and $(x_*, y_*) \in \mathcal{G}$, $l = 1, 2, 3$ in (3.16), we will apply the fourth-order NCE-RK method to the auxiliary ODE problem (3.1) starting from $t = t^n$ at each (x_*, y_*), where

$$y(t) = \begin{cases} u(x_*, y_*, t) \\ v(x_*, y_*, t) \\ w(x_*, y_*, t), \end{cases} \qquad F(t, y) = \begin{cases} -(f(u)_x + g(u)_y)|_{(x_*, y_*)} \\ -(p(u, v)_x + q(u, v)_y)|_{(x_*, y_*)} \\ -(h(u, w)_x + r(u, w)_y)|_{(x_*, y_*)}. \end{cases} \tag{3.17}$$

More specifically, one needs to evaluate the corresponding $K^{(i)}$, $i = 1, 2, 3, 4$ used in (3.2)–(3.3). In actual implementation, $F(t, y)$ is further replaced by

$$F(t, y) = \begin{cases} -(f'(u)v + g'(u)w)|_{(x_*, y_*)} \\ -(f''(u)v^2 + f'(u)\mathcal{R}_{v_x} + g''(u)wv + g'(u)\mathcal{R}_{v_y})|_{(x_*, y_*)} \\ -(f''(u)vw + f'(u)\mathcal{R}_{w_x} + g''(u)w^2 + g'(u)\mathcal{R}_{w_y})|_{(x_*, y_*)}, \end{cases} \tag{3.18}$$

and the operator $\mathcal{R}_{v_x}$ in (3.18) is to reconstruct v_x at (x_*, y_*), and it will be specified. Similar convention goes to $\mathcal{R}_{v_y}$, $\mathcal{R}_{w_x}$ and $\mathcal{R}_{w_y}$.

To evaluate $K^{(1)}$, we reconstruct the point value of $y(t^n)$, namely $p(x_*, y_*, t^n)$ $(p = u, v, w)$, based on the given cell averages $\{\bar{u}^n_{ij}, \bar{v}^n_{ij}, \bar{w}^n_{ij}\}_{ij}$ on the primal mesh. The operator $\mathcal{R}_q$ $(q = v_x, v_y, w_x, w_y)$ is chosen such that $q(x_*, y_*, t^n)$ is also reconstructed from the same set of cell averages. Such reconstructions are used in Section 2.2. Once $K^{(l)}$ for some $l \ge 1$ is available, then $Y^{(l+1)}$ can be computed based on (3.3). This will provide approximations for $p(x_*, y_*, t^n + c_{l+1}\Delta t)$ $(p = u, v, w)$. Now the operator $\mathcal{R}_q$ $(q = v_x, v_y, w_x, w_y)$ in (3.18) is chosen such that $q(x_*, y_*, t^n + c_{l+1}\Delta t)$ is reconstructed based on the point values $\{p(x_i, y_j, t^n + c_{l+1}\Delta t),\ p = u, v, w\}_{ij}$ using a fourth-order HWENO strategy similar to that in Section 2.2. With $K^{(l)}$, $l = 1, 2, 3, 4$, one can now follow the NCE-RK procedure to obtain accurate approximations for the point values of u, v, w in (3.16).

The algorithms discussed above can be extended to the system of conservation laws, see also Section 2.

4. Numerical examples

In this section, we will present a set of numerical experiments to illustrate the high-order accuracy and the robustness of the proposed methods to simulate one- and two-dimensional scalar or a system of hyperbolic conservation laws. The solutions can be either smooth or non-smooth. In one dimension, the numerical results are obtained by the proposed fifth-order central HWENO scheme with the third-order Lax–Wendroff method (C-HWENO5-LW3) or with the fourth-order NCE-RK method (C-HWENO5-NCERK4) in time. A uniform mesh with N elements is used. In two dimensions, the results are by the proposed fourth-order central HWENO scheme with the third-order Lax–Wendroff method (C-HWENO4-LW3) or with the fourth-order NCE-RK method (C-HWENO4-NCERK4) in time. A uniform mesh with $N_x \times N_y$ elements is used.

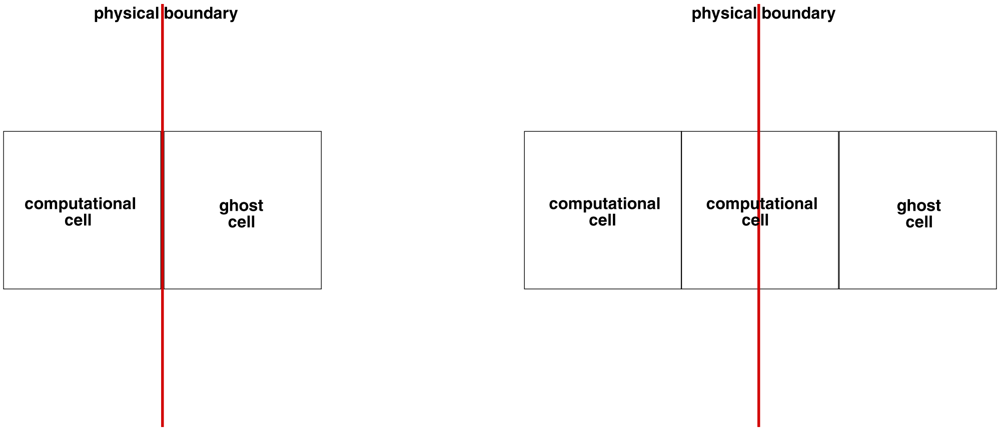

Fig. 2. Boundary cells for primal mesh (left) and dual mesh (right).

Table 1
The linear advection equation $u_t + u_x = 0$, with $u(x,0) = \sin(\pi x)$, and a periodic boundary condition. C-HWENO5-LW3 and C-HWENO5-NCERK4. $t = 2$. L_1 and L_∞ errors and orders of accuracy.

N	C-HWENO5-LW3				C-HWENO5-NCERK4			
	L_1 error	Order	L_∞ error	Order	L_1 error	Order	L_∞ error	Order
10	2.92e-001		4.83e-001		2.92e-001		4.83e-001	
20	1.85e-002	3.98	2.66e-002	4.18	1.85e-002	3.98	2.66e-002	4.18
40	6.85e-004	4.76	1.10e-003	4.60	6.85e-004	4.76	1.10e-003	4.60
80	2.11e-005	5.02	3.69e-005	4.90	2.11e-005	5.02	3.69e-005	4.90
160	6.34e-007	5.06	1.14e-006	5.02	6.34e-007	5.06	1.14e-006	5.02
320	1.89e-008	5.07	3.52e-008	5.02	1.89e-008	5.07	3.52e-008	5.02
640	5.60e-010	5.08	1.04e-009	5.08	5.60e-010	5.08	1.04e-009	5.08

The time step Δt is dynamically chosen. In particular, in one-dimensional scalar case, we take

$$\Delta t = C_{cfl} \frac{\Delta x}{\max|f'(u)|},$$

and in two-dimensional scalar case, we have

$$\Delta t = \frac{C_{cfl}}{(\max|f'(u)|/\Delta x + \max|g'(u)|/\Delta y)}.$$

The CFL number C_{cfl} is taken as 0.25 for one-dimensional tests, and 0.2 for two-dimensional ones except for some accuracy tests when the time step needs to be adjusted properly to ensure the spatial errors dominate. In the system case, $f'(u)$ and $g'(u)$ above are replaced by the eigenvalue of the Jacobian of $f(u)$ and $g(u)$, respectively, with the largest absolute value.

To apply boundary conditions, we introduce ghost cells for both the primal and dual meshes. For the primal mesh, the boundary of the physical domain coincides with the boundary of computational cells, and we define ghost cells outside of the physical domain (see left figure in Fig. 2). For the dual mesh, the boundary of the physical domain is located within the computational cells and ghost cells are introduced as shown in right figure of Fig. 2. The solutions in ghost cells are then given based on the boundary conditions, such as inflow/outflow, reflective, or periodic boundary conditions (see numerical examples below).

4.1. Accuracy tests with smooth solutions

We first validate the accuracy of the proposed schemes when the solutions are smooth. The conservation law can be linear, nonlinear, scalar, or system in one and two dimensions.

Example 4.1. We consider the one-dimensional linear advection equation,

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

with the initial condition $u(x,0) = \sin(\pi x)$, and a 2-periodic boundary condition. The schemes are run up to $t = 2$, and the CFL number $C_{cfl} = 0.01$. In Table 1 we present the L_1 and L_∞ errors and numerical orders of accuracy for C-HWENO5-LW3 and C-HWENO5-NCERK4 schemes. One can see that both schemes achieve their designed fifth-order accuracy with almost the same errors.

Table 2
Burgers equation $u_t + (u^2/2)_x = 0$, with $u(x,0) = 0.5 + \sin(\pi x)$, and a periodic boundary condition. C-HWENO5-LW3 and C-HWENO5-NCERK4. $t = 0.5/\pi$. L_1 and L_∞ errors and orders of accuracy.

N	C-HWENO5-LW3				C-HWENO5-NCERK4			
	L_1 error	Order	L_∞ error	Order	L_1 error	Order	L_∞ error	Order
10	1.19e-002		3.77e-002		1.19e-002		3.77e-002	
20	1.68e-003	2.82	1.21e-002	1.64	1.68e-003	2.82	1.21e-002	1.64
40	1.24e-004	3.76	1.10e-003	3.46	1.24e-004	3.76	1.10e-003	3.46
80	5.64e-006	4.46	5.18e-005	4.41	5.64e-006	4.46	5.18e-005	4.41
160	2.22e-007	4.67	2.11e-006	4.62	2.22e-007	4.67	2.09e-006	4.63
320	6.59e-009	5.07	7.69e-008	4.78	6.70e-009	5.05	7.48e-008	4.80
640	1.78e-010	5.21	1.50e-009	5.68	1.57e-010	5.42	1.61e-009	5.54

Table 3
Euler equations, with $\rho(x,0) = 1 + 0.2\sin(\pi x)$, $v(x,0) = 1$, $p(x,0) = 1$, and periodic boundary conditions. C-HWENO5-LW3 and C-HWENO5-NCERK4. $t = 2$. L_1 and L_∞ errors and orders of accuracy of density ρ.

N	C-HWENO5-LW3				C-HWENO5-NCERK4			
	L_1 error	Order	L_∞ error	Order	L_1 error	Order	L_∞ error	Order
10	1.75e-002		2.73e-002		1.75e-002		2.73e-002	
20	9.42e-004	4.22	1.47e-003	4.22	9.40e-004	4.22	1.47e-003	4.22
40	2.94e-005	5.00	5.42e-005	4.76	2.94e-005	5.00	5.42e-005	4.76
80	8.68e-007	5.08	1.67e-006	5.02	8.66e-007	5.09	1.67e-006	5.02
160	2.63e-008	5.04	5.18e-008	5.01	2.61e-008	5.05	5.15e-008	5.02
320	7.95e-010	5.05	1.43e-009	5.18	7.75e-010	5.07	1.40e-009	5.20
640	2.32e-011	5.10	3.83e-011	5.22	2.07e-011	5.23	3.50e-011	5.32

Example 4.2. We consider the one-dimensional Burgers equation, which is scalar and nonlinear,

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

with the initial condition $u(x,0) = 0.5 + \sin(\pi x)$, and a 2-periodic boundary condition. When $t = 0.5/\pi$ the solution is still smooth, and the L_1 and L_∞ errors and numerical orders of accuracy are presented in Table 2 for C-HWENO5-LW3 and C-HWENO5-NCERK4 schemes. We can see that both schemes achieve their designed fifth-order accuracy with comparable errors. We take the CFL number $C_{cfl} = 0.1$.

Example 4.3. We consider the one-dimensional nonlinear system of Euler equations

$$u_t + f(u)_x = 0 \tag{4.3}$$

with

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

Here ρ is the density, v is the velocity, E is the total energy, p is the pressure which is related to the conservative quantities through the relation $E = \frac{p}{\gamma-1} + \frac{1}{2}\rho v^2$, and $\gamma = 1.4$. The initial condition is set to be $\rho(x,0) = 1 + 0.2\sin(\pi x)$, $v(x,0) = 1$, $p(x,0) = 1$, with a 2-periodic boundary condition. The exact solution is $\rho(x,t) = 1 + 0.2\sin(\pi(x - t))$, $v(x,t) = 1$, and $p(x,t) = 1$. We compute the solution up to $t = 2$ with the CFL number $C_{cfl} = 0.1$. The L_1 and L_∞ errors and numerical orders of accuracy for the density ρ are shown in Table 3 for C-HWENO5-LW3 and C-HWENO5-NCERK4 schemes. Both schemes achieve their designed fifth-order accuracy with comparable errors.

Example 4.4. We consider the nonlinear Burgers equation in two dimensions

$$u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0 \tag{4.4}$$

with the initial condition $u(x,y,0) = 0.5 + \sin(\pi(x + y)/2)$, and a 4-periodic boundary condition in each direction. When $t = 0.5/\pi$ the solution is still smooth. The L_1 and L_∞ errors and numerical orders of accuracy are presented in Table 4 for C-HWENO4-LW3 and C-HWENO4-NCERK4 schemes, with $C_{cfl} = 0.2$. Both schemes achieve their designed fourth-order accuracy with comparable errors.

Example 4.5. We here consider the nonlinear system of Euler equations in two dimensions

$$U_t + f(U)_x + g(U)_y = 0 \tag{4.5}$$

Table 4
Burgers equation $u_t + (u^2/2)_x + (u^2/2)_y = 0$, with $u(x, y, 0) = 0.5 + \sin(\pi(x + y)/2)$, and periodic boundary conditions. C-HWENO4-LW3 and C-HWENO4-NCERK4. $t = 0.5/\pi$. L_1 and L_∞ errors and orders of accuracy.

$N_x \times N_y$	C-HWENO4-LW3				C-HWENO4-NCERK4			
	L_1 error	Order	L_∞ error	Order	L_1 error	Order	L_∞ error	Order
10 × 10	3.32e-002		8.33e-002		3.32e-002		8.33e-002	
20 × 20	3.60e-003	3.21	2.37e-002	1.81	3.60e-003	3.21	2.37e-002	1.81
40 × 40	4.02e-004	3.16	2.70e-003	3.13	4.02e-004	3.16	2.69e-003	3.13
80 × 80	2.09e-005	4.27	1.75e-004	3.95	2.09e-005	4.27	1.75e-004	3.95
160 × 160	1.10e-006	4.25	9.61e-006	4.19	1.10e-006	4.25	9.54e-006	4.20

Table 5
Euler equations, with $\rho(x, y, 0) = 1 + 0.2\sin(\pi(x + y))$, $u(x, y, 0) = 0.7$, $v(x, y, 0) = 0.3$, $p(x, y, 0) = 1$, and periodic boundary conditions. C-HWENO4-LW3 and C-HWENO4-NCERK4. $t = 2$. L_1 and L_∞ errors and orders of accuracy of density ρ.

$N_x \times N_y$	C-HWENO4-LW3				C-HWENO4-NCERK4			
	L_1 error	Order	L_∞ error	Order	L_1 error	Order	L_∞ error	Order
10 × 10	1.12e-001		1.72e-001		1.12e-001		1.72e-001	
20 × 20	6.44e-003	4.12	1.01e-002	4.09	6.65e-003	4.07	1.05e-002	4.03
40 × 40	1.95e-004	5.05	3.30e-004	4.94	1.93e-004	5.11	3.27e-004	5.01
80 × 80	7.05e-006	4.79	1.23e-005	4.75	6.85e-006	4.82	1.21e-005	4.76
160 × 160	2.49e-007	4.82	4.64e-007	4.73	2.34e-007	4.87	4.43e-007	4.77

with

$$U = (\rho, \rho u, \rho v, E)^T, \qquad f(U) = \big(\rho u, \rho u^2 + p, \rho u v, u(E + p)\big)^T,$$
$$g(U) = \big(\rho v, \rho u v, \rho v^2 + p, v(E + p)\big)^T.$$

Here ρ is the density, $(u, v)^T$ is the velocity, E is the total energy, p is the pressure which is related to the conservative quantities through the relation $E = \frac{p}{\gamma-1} + \frac{1}{2}\rho(u^2 + v^2)$, and $\gamma = 1.4$. 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 in each direction. The exact solution is $\rho(x, y, t) = 1 + 0.2\sin(\pi(x + y - (u + v)t))$, $u(x, y, t) = 0.7$, $v(x, y, t) = 0.3$, and $p(x, y, t) = 1$. We compute the solution up to $t = 2$ with the CFL number $C_{cfl} = 0.2$. The L_1 and L_∞ errors and numerical orders of accuracy of density ρ are reported in Table 5 for C-HWENO4-LW3 and C-HWENO4-NCERK4 schemes. Both schemes achieve their designed fourth or higher order accuracy with comparable errors.

4.2. Test cases with non-smooth solutions

We now test the performance of the proposed methods in terms of their resolution and non-oscillatory property when solving problems with non-smooth features, such as shocks, rarefaction, or contact discontinuity, in their solutions.

Example 4.6. We consider the same one-dimensional nonlinear Burgers equation (4.2) as in Example 4.2 with the same initial and boundary conditions, except that we now present the numerical solutions at $t = 1.5/\pi$ after a shock forms. In Fig. 3, the solutions of C-HWENO5-LW3 and C-HWENO5-NCERK4 schemes are plotted, together with the exact solution. The mesh is uniform with $N = 80$ elements. We can see that both schemes lead to non-oscillatory shock transitions in the solutions.

Example 4.7. We here consider a one-dimensional nonlinear non-convex scalar Buckley–Leverett problem

$$u_t + \left(\frac{4u^2}{4u^2 + (1 - u)^2}\right)_x = 0 \tag{4.6}$$

with the initial condition: $u = 1$ when $-\frac{1}{2} \le x \le 0$ and $u = 0$ elsewhere. The computational domain is $[-1, 1]$, with constant boundary conditions applied to both ends. The solution is computed up to $t = 0.4$. The exact solution contains shock, rarefaction, and a contact discontinuity. We remark that some high-order schemes may fail to converge to the correct entropy solution for this problem. In Fig. 4, the solutions of C-HWENO5-LW3 and C-HWENO5-NCERK4 schemes are shown with $N = 80$ mesh elements, together with the exact solution. One can see both schemes capture the correct entropy solution well, with good resolutions for all the major features in the solution.

Example 4.8. We solve the one-dimensional Euler equations (4.3) with a Riemann initial condition for the Lax problem

$$(\rho, v, p) = (0.445, 0.698, 3.528) \quad \text{for } x \le 0, \qquad (\rho, v, p) = (0.5, 0, 0.571) \quad \text{for } x > 0.$$

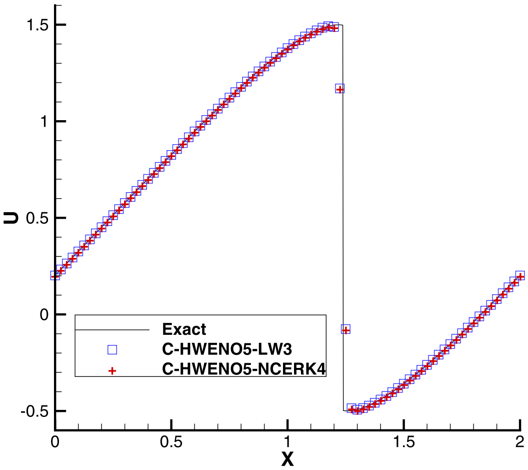

Fig. 3. Burgers equation in one dimension. $u(x,0) = 0.5 + \sin(\pi x)$. $t = 1.5/\pi$ and $N = 80$. Solid line: exact solution; square: C-HWENO5-LW3; plus: C-HWENO5-NCERK4.

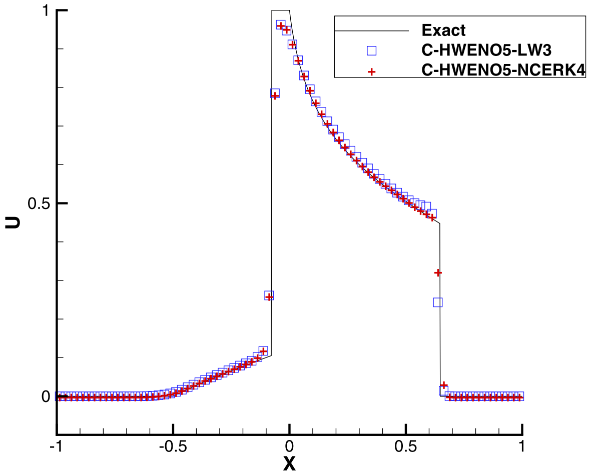

Fig. 4. The Buckley–Leverett problem. $t = 0.4$ and $N = 80$. Solid line: exact solution; square: C-HWENO5-LW3; plus: C-HWENO5-NCERK4.

The computational domain is $[-5, 5]$, with inflow/outflow boundary conditions applied to left/right ends. The schemes are run up to $t = 1.3$. In Fig. 5, we plot the computed density ρ with $N = 200$ mesh elements by C-HWENO5-LW3 and C-HWENO5-NCERK4 schemes, together with the exact solution. One can see that both schemes render equally good non-oscillatory shock transitions for this problem.

Remark 3. For the Riemann problems such as Example 4.8, and the next two Examples 4.9 and 4.10, the initial cell average of the derivative of the solution is assigned as

$$\bar{v}_i^0 = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} v(x,0)dx = \frac{1}{\Delta x}\big(u_0(x_{i+1/2}) - u_0(x_{i-1/2})\big).$$

Example 4.9. The non-smooth examples we have presented so far contain simple smooth regions in the solutions, for which capturing the shock sharply with non-oscillatory transitions is the main focus and usually a good second-order non-oscillatory scheme would give satisfactory results. The examples presented here are mainly to demonstrate the non-oscillatory properties of the proposed high-order schemes. A high-order scheme would be more advantageous when the solution contains both shocks and complex smooth structures. One such example is the Shu–Osher problem, which describes shock interacting with entropy waves [30]. This example is modeled by the one-dimensional Euler equations (4.3) with a moving Mach-3 shock interacting with sine waves in density, i.e. initially

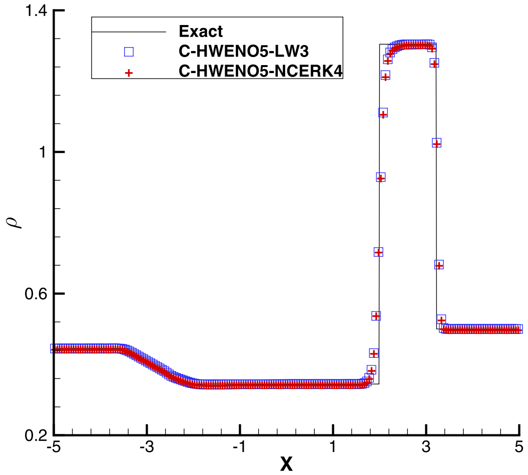

Fig. 5. The Lax problem. $t = 1.3$ and $N = 200$. Density ρ. Solid line: exact solution; square: C-HWENO5-LW3; plus: C-HWENO5-NCERK4.

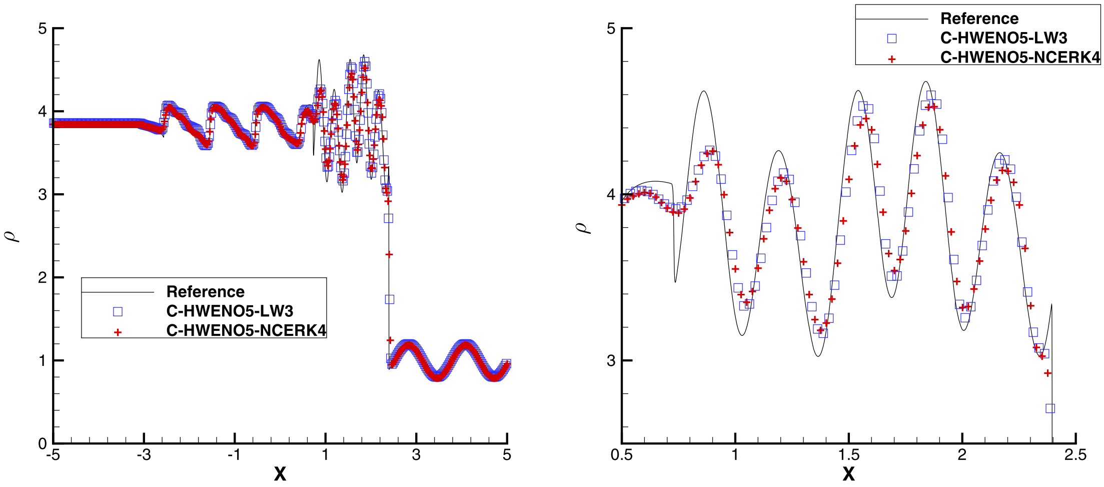

Fig. 6. The shock density wave interaction problem. $t = 1.8$ and $N = 400$. Density ρ. Solid line: reference solution; square: C-HWENO5-LW3; plus: C-HWENO5-NCERK4.

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

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

The computational domain is $[-5, 5]$, and the boundary conditions are taken to the same as the initial data. Here we take $\sigma = 0.2$. In Fig. 6, we plot the computed density ρ at $t = 1.8$ by C-HWENO5-LW3 and C-HWENO5-NCERK4 schemes with $N = 400$ mesh elements, together with a reference solution obtained by the fifth-order finite difference WENO scheme [9] with 16 000 grid points, we further zoom in the oscillatory region of the solution between $x = 0.5$ and $x = 2.5$. One can see that this example very well demonstrates the good resolution of the proposed methods to resolve smooth features in the solution as well as the non-oscillatory nature of the methods to capture discontinuities.

Example 4.10. We here consider the interaction of blast waves, modeled by the one-dimensional Euler equations (4.3) with the initial condition

$$(\rho, v, p) = (1, 0, 1000) \quad \text{for } 0 \leq x < 0.1,$$

$$(\rho, v, p) = (1, 0, 0.01) \quad \text{for } 0.1 \leq x < 0.9,$$

$$(\rho, v, p) = (1, 0, 100) \quad \text{for } 0.9 \leq x.$$

The computational domain is $[0, 1]$, with reflecting boundary conditions applied to both ends, see [7,32]. In Fig. 7, we plot the computed density ρ at $t = 0.038$ by C-HWENO5-LW3 and C-HWENO5-NCERK4 schemes with $N = 800$ mesh elements, together with a reference solution obtained by the fifth-order finite difference WENO scheme [9] with 16 000 grid points. We can see that both schemes give equally good resolution for this problem.

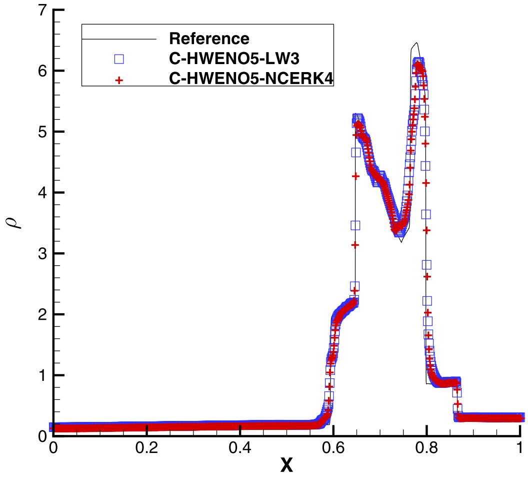

Fig. 7. The interaction of blast waves problem. $t = 0.038$ and $N = 800$. Density ρ. Solid line: reference solution; square: C-HWENO5-LW3; plus: C-HWENO5-NCERK4.

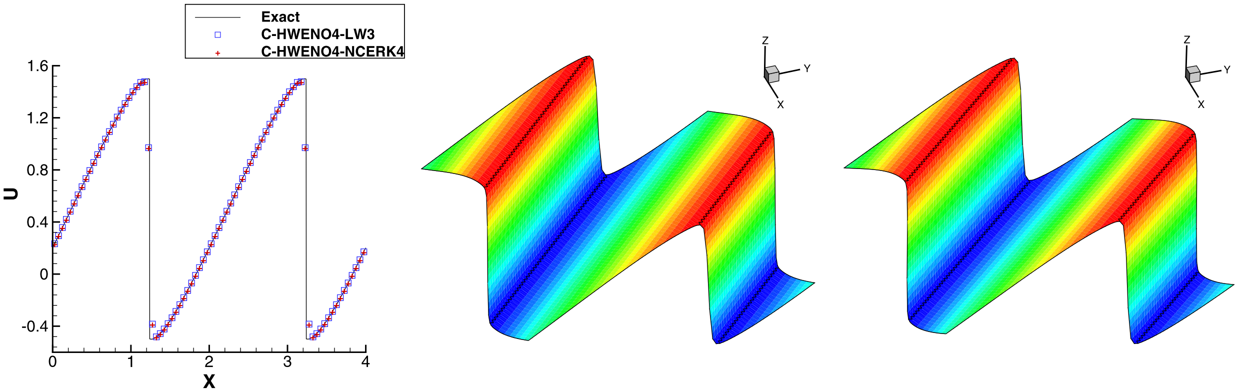

Fig. 8. Burgers equation in two dimensions. $u(x, y, 0) = 0.5 + \sin(\pi(x + y)/2)$. $t = 1.5/\pi$ and $N_x \times N_y = 80 \times 80$. A slide of the solution (left) at $x = y$, solid line: exact solution; square: C-HWENO4-LW3; plus: C-HWENO4-NCERK4. The surface of solution computed by C-HWENO4-LW3 (middle) and C-HWENO4-NCERK4 (right).

Example 4.11. We solve the same two-dimensional Burgers equation (4.4) as in Example 4.4 with the same initial and boundary conditions, except that we now present the numerical solutions at $t = 1.5/\pi$ after a shock forms. In Fig. 8, we plot a slice of the computed solution at $x = y$ on an 80×80 mesh by C-HWENO4-LW3 and C-HWENO4-NCERK4 schemes, together with the exact solution. The surfaces of the solutions computed by C-HWENO4-LW3 and C-HWENO4-NCERK4 schemes are also shown. Both schemes render good non-oscillatory shock transitions for this problem.

Example 4.12. We here consider the example of double Mach reflection, which is originally from [32] and modeled by the two-dimensional Euler equations (4.5). The computational domain is $[0, 4] \times [0, 1]$. The reflecting wall lies at the bottom, 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 x-axis. For the bottom boundary, the exact post-shock 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, the flow values are set to describe the exact motion of a Mach-10 shock. Post-shock and pre-shock conditions are imposed for the left and right boundary conditions, respectively. We compute up to $t = 0.2$. We use three different uniform meshes, with 240×60, 480×120, 960×240 mesh elements. We only show the results with 960×240 mesh elements to save space. In Fig. 9, the density contour plots are presented in the region of $[0, 3] \times [0, 1]$ by C-HWENO4-LW3 and C-HWENO4-NCERK4 schemes, respectively. Both schemes show good resolution to approximate the solution with comparable results. All contour plots are with 30 equally-spaced contour lines for the density from 1.5 to 22.7.

Example 4.13. Our final example is about a Mach-3 wind tunnel with a step. It is from [32] and modeled by the two-dimensional Euler equations (4.5). The setup of the problem is as follows. The wind tunnel is 1 length unit wide and 3

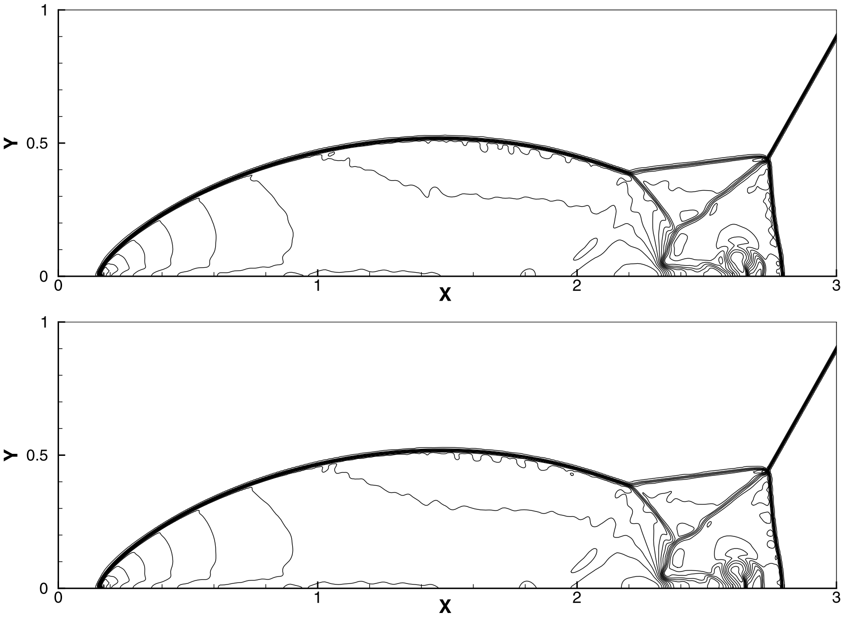

Fig. 9. Double Mach reflection problem. $t = 0.2$ and $N_x \times N_y = 960 \times 240$. C-HWENO4-LW3 (top) and C-HWENO4-NCERK4 (bottom). 30 equally spaced density contours from 1.5 to 22.7.

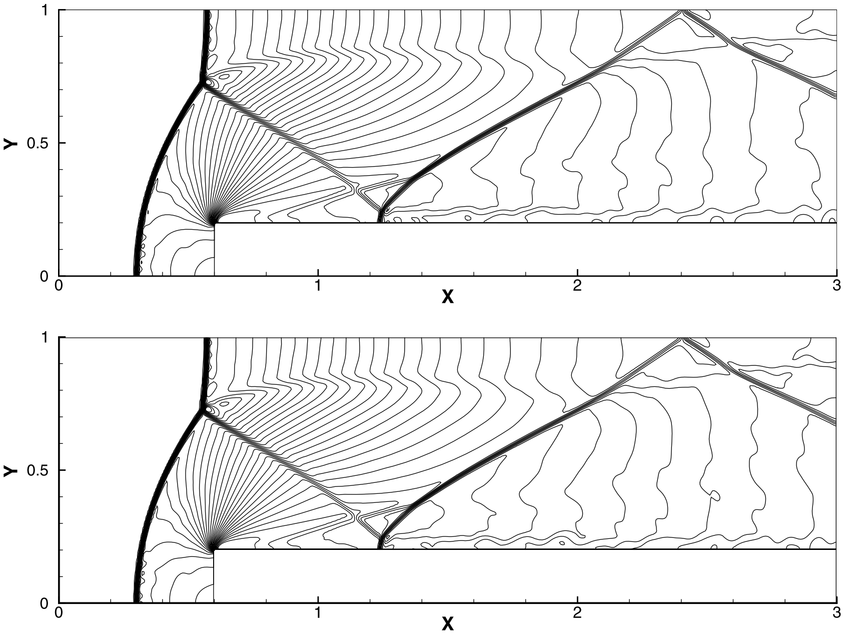

Fig. 10. Forward step problem. $t = 4$ and $N_x \times N_y = 480 \times 160$. C-HWENO4-LW3 (top) and C-HWENO4-NCERK4 (bottom). 30 equally spaced density contours from 0.54 to 6.15.

length units long. The step is 0.2 length units high and is located 0.6 length units from the left end of the tunnel. The problem is initialized by a right-going Mach-3 flow. Reflective boundary conditions are applied along the walls of the tunnel, and inflow/outflow boundary conditions are applied at the entrance/exit. The corner of the step is a singular point and we treat it based on the assumption of a nearly steady flow in the region near the corner. We compute up to $t = 4$. We use three different uniform meshes, with 120×40, 240×80, 480×160 mesh elements. Again, we only show the results with 480×160 mesh elements to save space. In Fig. 10, the density contour plots are presented by C-HWENO4-LW3 and C-HWENO4-NCERK4 schemes, respectively. Both schemes show good resolution to approximate the solution with comparable results. All contour plots are with 30 equally-spaced contour lines for the density from 0.54 to 6.15.

Remark 4. For both Examples 4.12 and 4.13, the initial cell averages of the derivatives of the solutions are simply taken as zero, namely, $\bar{v}^0_{ij} = 0$, $\overline{w}^0_{ij} = 0$.

5. Concluding remarks

In this paper, we design high-order central Hermite WENO schemes for solving one- and two-dimensional linear and nonlinear hyperbolic conservation laws. The methods use Hermite WENO reconstruction as spatial discretizations, Lax–Wendroff type methods or the natural continuous extension of Runge–Kutta methods as time discretizations in a central finite volume framework on staggered meshes. Our schemes have the advantages of being compact in reconstruction and requiring no flux splitting or numerical fluxes. In the system case, local characteristic decomposition is applied in the reconstruction of cell averages to achieve better non-oscillatory properties of the schemes.

The spatial reconstructions proposed in this work can be directly applied to the central finite volume schemes on two overlapping meshes [20], that evolve two sets of numerical approximations for cell averages of the solution and its first derivative(s), defined on both meshes at the same discrete times. Central schemes on staggered meshes as in our current work are computationally more efficient due to the halved total number of degrees of freedom. On the other hand, central schemes on two overlapping meshes were introduced to reduce the large numerical dissipation of the central schemes on staggered meshes when the time step size is very small compared to the spatial mesh size, and they can also be formulated as semi-discrete methods hence can be easily combined with the commonly used nonlinearly stable TVD Runge–Kutta time discretizations [29] following method of lines approaches.

Acknowledgements

Zhanjing Tao finished part of the work during his visit to the Department of Mathematical Sciences of Rensselaer Polytechnic Institute during January–July, 2014. He appreciates the support and hospitality of the Department, and also wants to thank Graduate School of Xiamen University for sponsoring this visit through the Short-term Research Program.

Appendix A

In two-dimensional case, one can use the governing equations (2.21)–(2.23) to convert time derivatives, up to the third-order accuracy, of u, v and w into spatial ones as below.

$$\begin{aligned}
u_t &= -f(u)_x - g(u)_y = -\big(f'(u)v + g'(u)w\big),\\
u_{tt} &= -\big(f'(u)u_t\big)_x - \big(g'(u)u_t\big)_y = -\big(f''(u)vu_t + f'(u)u_{xt} + g''(u)wu_t + g'(u)u_{yt}\big),\\
u_{xt} &= -\big(f''(u)(v)^2 + f'(u)v_x + g''(u)vw + g'(u)w_x\big),\\
u_{yt} &= -\big(f''(u)wv + f'(u)v_y + g''(u)(w)^2 + g'(u)w_y\big),\\
u_{ttt} &= -\big(f''(u)(u_t)^2 + f'(u)u_{tt}\big)_x - \big(g''(u)(u_t)^2 + g'(u)u_{tt}\big)_y,\\
v_t &= -\big(f'(u)v\big)_x - \big(g'(u)v\big)_y = -\big(f''(u)(v)^2 + f'(u)v_x + g''(u)wv + g'(u)v_y\big),\\
v_{tt} &= -\big(f''(u)u_tv + f'(u)v_t\big)_x - \big(g''(u)u_tv + g'(u)v_t\big)_y\\
&= -\big(f'''(u)vu_tv + f''(u)(u_{xt}v + u_tv_x + vv_t) + f'(u)v_{xt}\big)\\
&\quad - \big(g'''(u)wu_tv + g''(u)(u_{yt}v + u_tv_y + wv_t) + g'(u)v_{yt}\big),\\
v_{xt} &= -\big(f''(u)vv + f'(u)v_x + g''(u)wv + g'(u)v_y\big)_x\\
&= -\big(f'''(u)(v)^3 + f''(u)(v_xv + 2vv_x) + f'(u)v_{xx}\big)\\
&\quad - \big(g'''(u)vwv + g''(u)(w_xv + wv_x + vv_y) + g'(u)v_{xy}\big),\\
v_{yt} &= -\big(f''(u)vv + f'(u)v_x + g''(u)wv + g'(u)v_y\big)_y\\
&= -\big(f'''(u)w(v)^2 + f''(u)(v_yv + vv_y + wv_x) + f'(u)v_{xy}\big)\\
&\quad - \big(g'''(u)(w)^2v + g''(u)(w_yv + 2wv_y) + g'(u)v_{yy}\big),\\
v_{ttt} &= -\big(f'''(u)(u_t)^2v + f''(u)(u_{tt}v + 2u_tv_t) + f'(u)v_{tt}\big)_x\\
&\quad - \big(g'''(u)(u_t)^2v + g''(u)(u_{tt}v + 2u_tv_t) + g'(u)v_{tt}\big)_y,\\
w_t &= -\big(f'(u)w\big)_x - \big(g'(u)w\big)_y = -\big(f''(u)vw + f'(u)w_x + g''(u)(w)^2 + g'(u)w_y\big),
\end{aligned}$$

$$\begin{aligned}
w_{tt} &= -\big(f''(u)u_t w + f'(u)w_t\big)_x - \big(g''(u)u_t w + g'(u)w_t\big)_y \\
&= -\big(f'''(u)v u_t w + f''(u)(u_{xt}w + u_t w_x + v w_t) + f'(u)w_{xt}\big) \\
&\quad - \big(g'''(u)w u_t w + g''(u)(u_{yt}w + u_t w_y + w w_t) + g'(u)w_{yt}\big), \\
w_{xt} &= -\big(f''(u)v w + f'(u)w_x + g''(u)w w + g'(u)w_y\big)_x \\
&= -\big(f'''(u)(v)^2 w + f''(u)(v_x w + 2v w_x) + f'(u)w_{xx}\big) \\
&\quad - \big(g'''(u)v(w)^2 + g''(u)(w_x w + w w_x + v w_y) + g'(u)w_{xy}\big), \\
w_{yt} &= -\big(f''(u)v w + f'(u)w_x + g''(u)w w + g'(u)w_y\big)_y \\
&= -\big(f'''(u)w v w + f''(u)(v_y w + v w_y + w w_x) + f'(u)w_{xy}\big) \\
&\quad - \big(g'''(u)(w)^3 + g''(u)(w_y w + 2w w_y) + g'(u)w_{yy}\big), \\
w_{ttt} &= -\big(f'''(u)(u_t)^2 w + f''(u)(u_{tt}w + 2u_t w_t) + f'(u)w_{tt}\big)_x \\
&\quad - \big(g'''(u)(u_t)^2 w + g''(u)(u_{tt}w + 2u_t w_t) + g'(u)w_{tt}\big)_y.
\end{aligned}$$

References

[1] P. Arminjon, D. Stanescu, M.-C. Viallon, A two-dimensional finite volume extension of the Lax–Friedrichs and Nessyahu–Tadmor schemes for compressible flow, in: M. Hafez, K. Oshima (Eds.), Proceedings of the 6th International Symposium on Computational Fluid Dynamics, vol. IV, Lake Tahoe, NV, 1995, pp. 7–14.

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

[3] F. Bianco, G. Puppo, G. Russo, High-order central schemes for hyperbolic systems of conservation laws, SIAM J. Sci. Comput. 21 (1999) 294–322.

[4] R.L. Dougherty, A.S. Edelman, J.M. Hyman, Nonnegativity-, monotonicity-, or convexity-preserving cubic and quintic Hermite interpolation, Math. Comput. 52 (1989) 471–494.

[5] O. Friedrichs, Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids, J. Comput. Phys. 144 (1998) 194–212.

[6] O. Friedrichs, P.D. Lax, Systems of conservation equations with a convex extension, Proc. Natl. Acad. Sci. USA 68 (1971) 1686–1688.

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

[8] C. Hu, C.-W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys. 150 (1999) 97–127.

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

[10] G. Jiang, E. Tadmor, Nonoscillatory central schemes for multidimensional hyperbolic conservation laws, SIAM J. Sci. Comput. 19 (1998) 1892–1917.

[11] A. Kurganov, D. Levy, A third-order semidiscrete central scheme for conservation laws and convection–diffusion equations, SIAM J. Sci. Comput. 22 (2000) 1461–1488.

[12] A. Kurganov, G. Petrova, A third-order semi-discrete genuinely multidimensional central scheme for hyperbolic conservation laws and related problems, Numer. Math. 88 (2001) 683–729.

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

[14] D. Levy, G. Puppo, G. Russo, Central WENO schemes for hyperbolic systems of conservation laws, Math. Model. Numer. Anal. 33 (1999) 547–571.

[15] D. Levy, G. Puppo, G. Russo, Compact central WENO schemes for multidimensional conservation laws, SIAM J. Sci. Comput. 22 (2000) 656–672.

[16] D. Levy, G. Puppo, G. Russo, A third order central WENO scheme for 2D conservation laws, Appl. Numer. Math. 33 (2000) 415–421.

[17] D. Levy, G. Puppo, G. Russo, A fourth-order central WENO scheme for multidimensional hyperbolic systems of conservation laws, SIAM J. Sci. Comput. 24 (2002) 480–506.

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

[19] X. Liu, E. Tadmor, Third order nonoscillatory central scheme for hyperbolic conservation laws, Numer. Math. 79 (1998) 397–425.

[20] Y. Liu, Central schemes on overlapping cells, J. Comput. Phys. 209 (2005) 82–104.

[21] T. Nakamura, R. Tanaka, T. Yabe, K. Takizawa, Exactly conservative semi-Lagrangian scheme for multi-dimensional hyperbolic equations with directional splitting technique, J. Comput. Phys. 174 (2001) 171–207.

[22] H. Nessyahu, E. Tadmor, Non-oscillatory central differencing for hyperbolic conservation laws, J. Comput. Phys. 87 (1990) 408–463.

[23] J. Qiu, C.-W. Shu, On the construction, comparison, local characteristic decomposition for high order central WENO schemes, J. Comput. Phys. 183 (2002) 187–209.

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

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

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

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

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

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

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

[31] H. Takewaki, A. Nishiguchi, T. Yabe, Cubic interpolated pseudoparticle method (CIP) for solving hyperbolic type equations, J. Comput. Phys. 61 (1985) 261–268.

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

[33] M. Zennaro, Natural continuous extensions of Runge–Kutta methods, Math. Comput. 46 (1986) 119–133.

[34] J. Zhu, J. Qiu, A class of the fourth order finite volume Hermite weighted essentially non-oscillatory schemes, Sci. China Ser. A 51 (2008) 1549–1560.