LOCAL CHARACTERISTIC DECOMPOSITION FREE HIGH ORDER FINITE DIFFERENCE WENO SCHEMES FOR HYPERBOLIC SYSTEMS ENDOWED WITH A COORDINATE SYSTEM OF RIEMANN INVARIANTS*

ZIYAO XU† AND CHI-WANG SHU‡

Abstract. The weighted essentially non-oscillatory (WENO) schemes are popular high order numerical methods for hyperbolic conservation laws. When dealing with hyperbolic systems, WENO schemes are usually used in cooperation with the local characteristic decomposition, as the component-wise WENO reconstruction/interpolation procedure often produces oscillatory approximations near shocks. In this paper, we investigate local characteristic decomposition free WENO schemes for a special class of hyperbolic systems endowed with a coordinate system of Riemann invariants. We apply the WENO procedure to the coordinate system of Riemann invariants instead of the local characteristic fields to save the expensive computational cost on local characteristic decomposition but meanwhile maintain the essentially non-oscillatory performance. Due to the nonlinear algebraic relation between the Riemann invariants and conserved variables, it is difficult to obtain the cell averages of Riemann invariants directly from those of conserved variables, and vice versa, thus we do not use the finite volume WENO schemes in this work. The same difficulty is also faced in the traditional Shu-Osher lemma [25] based finite difference schemes, as the computation of fluxes is based on reconstruction as well. Therefore, we adopt the alternative formulation of finite difference WENO scheme [13, 24] in this paper, which is based on interpolation for nodal values. The efficiency and good performance of our method are demonstrated by extensive numerical tests, which indicate the coordinate system of Riemann invariants is a good alternative of local characteristic fields for the WENO procedure.

Key words. hyperbolic systems, coordinate system of Riemann invariants, alternative formulation of finite difference WENO schemes, local characteristic decomposition free

MSC codes. 65M06

1. Introduction.

It has long been recognized that, the solutions of nonlinear hyperbolic equations can develop discontinuities (shocks) in finite time, even if the initial condition is smooth. Such a phenomenon greatly challenges the robustness of high order numerical methods, as spurious oscillations typically appear near shocks in numerical approximations (the Gibbs phenomenon), and may blow/mess up the simulation in later times. There have been numerous high order numerical methods developed to address this issue, among which the essentially non-oscillatory (ENO)/weighted essentially non-oscillatory (WENO) schemes have gained great success and have been widely used in applications.

The ENO methods, first developed by Harten et al. [10], use adaptive strategy to choose the smoothest stencil among several candidates to reconstruct the solution from its cell averages, hence the methods yield essentially non-oscillatory approximation near shocks. The original ENO scheme was based on the framework of finite volume methods, where the numerical fluxes at cell interfaces are obtained through reconstructed solution. Later, Shu and Osher proposed the finite difference ENO scheme in [24] based on ENO interpolation for nodal values and high order finite

*Submitted to the editors DATE.

Funding: Research supported by AFOSR grant FA9550-20-1-0055 and NSF grant DMS-2010107.

†Division of Applied Mathematics, Brown University, Providence, RI 02912, USA. (ziyao_xu1@brown.edu).

‡Division of Applied Mathematics, Brown University, Providence, RI 02912, USA. (chi-wang_shu@brown.edu).

difference approximation for spatial derivatives of fluxes, which saves considerable computational cost in multi-dimensions, as the derivatives can be approximated dimension by dimension in finite difference schemes. Their subsequent work in [25] developed a simpler finite difference ENO scheme based on the Shu-Osher lemma to approximate the fluxes at cell interfaces by standard reconstruction for fluxes at grid points. The WENO methods were developed upon ENO, with the idea of using a convex combination of all candidate stencils rather than only one stencil in the original ENO scheme. In the pioneer work of WENO schemes, Liu et al. [14] used linear weights to combine the candidate stencils in r-th order ENO schemes to yield $(r+1)$-th order of accuracy. It was later improved by Jiang and Shu [12] to achieve $(2r-1)$-th order of accuracy on the same stencils, by adopting nonlinear weights based on smoothness indicators designed for optimal accuracy in smooth regions and essentially non-oscillatory fashion near discontinuities. Thereafter, intensive modifications and improvements of the WENO procedure have been developed, e.g. the mapped WENO [11], WENO-Z [4, 6], modified WENO to handle negative weights [21], multi-resolution WENO [29], Hermite WENO [19], among other variants. Both finite volume [10] and finite difference [24, 25] frameworks for ENO can be used with the above WENO procedures. In our work, we use the classic WENO-JS procedure [12], as it is most widely used and relatively simple to code. For more details about the history and development of ENO and WENO methods, one can refer to the surveys [22, 23].

The ENO/WENO methods perform very well for scalar conservation laws as they achieve uniformly high order accuracy in smooth regions and resolve shocks sharply with essentially non-oscillatory quality. However, when dealing with hyperbolic systems, the component-wise ENO/WENO procedure often produces oscillatory results near shocks, especially when waves corresponding to different characteristic fields interact, such as in Riemann problems. The primary approach to resolve this problem is to apply the ENO/WENO procedure to the local characteristic fields of the system obtained by local characteristic decomposition for the conserved variables/fluxes, and transform the results back to the conserved variables/fluxes afterwards. Below, we briefly review how the WENO methods for hyperbolic systems are used in cooperation with the local characteristic decomposition. For the ease of comparison with the algorithm to be developed in this paper, we demonstrate it as per example of the alternative formulation of finite difference WENO scheme developed in [13] from [24], which will be introduced with more details in Section 3.

We consider the hyperbolic system of m ($m>1$) components

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = \mathbf{0}, \tag{1.1}$$

in one space dimension, where $\mathbf{u} = (u_1, \ldots, u_m) \in \mathbb{R}^m$ are the conserved variables and $\mathbf{f}(\mathbf{u}) = (f_1(\mathbf{u}), \ldots, f_m(\mathbf{u})) \in \mathbb{R}^m$ are the fluxes. Now and henceforth, we use bold face font to denote vectors or matrices.

Consider uniform grids with the grid point $x_j = j\Delta x$ centering in the cell $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}] = [(j-\frac{1}{2})\Delta x, (j+\frac{1}{2})\Delta x], \forall j \in \mathbb{Z}$. The semi-discrete $(2r-1)$-th order alternative formulation of finite difference WENO scheme for (1.1) is formulated as

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

where $\mathbf{u}_j$ is the approximation to $\mathbf{u}(x_j, t)$, $\hat{\mathbf{f}}_{j+\frac{1}{2}} = \hat{\mathbf{f}}(\mathbf{u}^-_{j+\frac{1}{2}}, \mathbf{u}^+_{j+\frac{1}{2}}, \cdots)$ is the numerical flux, whose definition and arguments omitted for brevity will be detailed

in later sections, and $\mathbf{u}^{\pm}_{j+\frac{1}{2}}$ are approximations to $\mathbf{u}(x_{j+\frac{1}{2}},t)$ from interpolants on I_j and I_{j+1}. We denote the WENO interpolation for a scalar-valued grid function v at $x_{j+\frac{1}{2}}$ on I_j by $v^-_{j+\frac{1}{2}} = \text{weno}(v_{j-r+1},\ldots,v_{j+r-1})$, whose implementation will be detailed in Section 3. The WENO interpolation for $v^+_{j-\frac{1}{2}}$ follows from mirror symmetry, i.e. $v^+_{j-\frac{1}{2}} = \text{weno}(v_{j+r-1},\ldots,v_{j-r+1})$. We shall abuse the notation to also let it denote the component-wise WENO interpolation for vectors, e.g. $\mathbf{v}^-_{j+\frac{1}{2}} = \text{weno}(\mathbf{v}_{j-r+1},\ldots,\mathbf{v}_{j+r-1})$.

The flowchart of the alternative formulation of finite difference WENO algorithm (1.2) with local characteristic decomposition, based on the nodal values $\{\mathbf{u}^n_j\}_{j\in\mathbb{Z}}$ at time level t^n, is given as follows, where the superscript n is omitted for brevity and the computation is carried out for all $j\in\mathbb{Z}$:

1. Approximate the solution at $x_{j+\frac{1}{2}}$ by the arithmetic mean $\mathbf{u}_{j+\frac{1}{2}} = \frac{1}{2}(\mathbf{u}_j+\mathbf{u}_{j+1})$, or the Roe's average [20] satisfying $\mathbf{f}(\mathbf{u}_{j+1})-\mathbf{f}(\mathbf{u}_j) = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{u}_{j+\frac{1}{2}})(\mathbf{u}_{j+1}-\mathbf{u}_j)$, if it is available.
2. Perform the eigendecomposition on the Jacobian matrix: $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{u}_{j+\frac{1}{2}}) = \mathbf{R}_{j+\frac{1}{2}}\mathbf{\Lambda}_{j+\frac{1}{2}}\mathbf{R}^{-1}_{j+\frac{1}{2}}$, where $\mathbf{\Lambda}_{j+\frac{1}{2}}$ and $\mathbf{R}_{j+\frac{1}{2}}$ are the diagonal matrix containing all eigenvalues and the eigenmatrix consist of a complete set of eigenvectors as its columns, respectively, of the Jacobian matrix.
3. Calculate the local characteristic variables: $\mathbf{v}_i = \mathbf{R}^{-1}_{j+\frac{1}{2}}\mathbf{u}_i$, on the stencils $i = j-r+1,\ldots,j+r$.
4. Perform the WENO interpolation for the local characteristic variables to obtain $\mathbf{v}^-_{j+\frac{1}{2}} = \text{weno}(\mathbf{v}_{j-r+1},\ldots,\mathbf{v}_{j+r-1})$ and $\mathbf{v}^+_{j+\frac{1}{2}} = \text{weno}(\mathbf{v}_{j+r},\ldots,\mathbf{v}_{j-r+2})$.
5. Transform the local characteristic variables back to the conserved variables: $\mathbf{u}^{\pm}_{j+\frac{1}{2}} = \mathbf{R}_{j+\frac{1}{2}}\mathbf{v}^{\pm}_{j+\frac{1}{2}}$.
6. Calculate the numerical fluxes $\hat{\mathbf{f}}_{j+\frac{1}{2}}$ to evolve the scheme (1.2) in time.

As we can see, the steps 1, 2, 3 and 5 are extra costs due to the local characteristic decomposition. In particular, there are $2r$ matrix-vector multiplications at every cell interface $x_{j+\frac{1}{2}}$ at the step 3, which is responsible for most of the floating point operations.

There have been some attempts on avoiding or reducing the costs on local characteristic decomposition in numerical schemes, meanwhile maintaining the essentially non-oscillatory performance, but only limited successes were achieved. In [12], Jiang and Shu computed the weights in WENO from entropy and pressure instead of the characteristic variables for Euler systems, to reduce part of the operations in local characteristic decomposition. In [28], Zheng et al. argued that at the contact discontinuity on interface of two-medium flow, direct WENO interpolation for primary variables is better than component-wise interpolation for conserved variables, but local characteristic decomposition was still applied therein to the primitive variables to get more satisfactory results. Low order central schemes [16, 15] can be used without local characteristic decomposition. However, the local characteristic decomposition is still necessary to control spurious oscillations when orders of the schemes are high [17].

In this work, we propose an efficient implementation of finite difference WENO schemes that is local characteristic decomposition free, for a special class of hyperbolic systems endowed with a coordinate system of Riemann invariants. Examples of such systems include all two-component hyperbolic systems and some multi-component systems to be introduced in Section 2. The key idea of the method is to apply the

WENO procedure to the nodal values of the coordinate system of Riemann invariants, which are (one-to-one) nonlinear algebraic functions of the conserved variables, and transform the interpolated values back to the conserved variables in the calculation of fluxes. The improvement in efficiency is due to the fact that, the characteristic decomposition for the WENO procedure is calculated locally, namely the conserved variables/fluxes at every node need to be projected onto local characteristic fields by different inverse eigenmatrices at different cell interfaces, while the Riemann invariants have definite algebraic relation with the conserved variables thus only need to be calculated once per node. A comparison of floating point operations in these two methods are shown in Appendix A. The good non-oscillatory performance of such treatment is justified by both theoretical properties of hyperbolic systems and numerical tests.

Due to the nonlinearity of the algebraic relation between Riemann invariants and conserved variables/fluxes, we cannot use any reconstruction based numerical schemes like the finite volume WENO or the traditional Shu-Osher lemma based finite difference WENO, as we cannot directly transfer the cell averages between Riemann invariants and conserved variables/fluxes. On the other hand, the transform between nodal values is straightforward, thus we adopt the alternative formulation of finite difference WENO scheme [13], which is based on WENO interpolation for nodal values. Its implementation will be demonstrated in Section 3. For detailed introduction and comparison with the traditional finite difference WENO for the alternative formulation, one can refer to [13].

The rest of the paper is organized as follows. In Section 2, we review the definition of Riemann invariants and their important properties, and give examples of hyperbolic systems endowed with a coordinate system of Riemann invariants. In Section 3, we give a detailed description for our algorithm. We use numerical tests in Section 4 to demonstrate the efficiency and good performance of our methods. Finally, we end up with some concluding remarks in Section 5.

2. Riemann invariants.

In this section, we review the definition and important properties of Riemann invariants of hyperbolic system of conservation laws.

We consider the hyperbolic system (1.1), with $\mathbf{u} = (u_1, \dots, u_m)^T$ the conserved variables taking values in an open set $\mathcal{O} \subset \mathbb{R}^m$, and $\mathbf{f}(\mathbf{u}) = (f_1(\mathbf{u}), \dots, f_m(\mathbf{u}))^T$ a smooth flux function on $\mathcal{O}$. From hyperbolicity, the Jacobian matrix $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$ has a complete set of eigenvectors $\mathbf{r}_1(\mathbf{u}), \mathbf{r}_2(\mathbf{u}), \dots, \mathbf{r}_m(\mathbf{u})$ corresponding to the real eigenvalues $\lambda_1(\mathbf{u}) \le \lambda_2(\mathbf{u}) \le \ldots \le \lambda_m(\mathbf{u})$, for all $\mathbf{u} \in \mathcal{O}$.

The Riemann invariants of the hyperbolic system (1.1) is defined as follows [26]:

DEFINITION 2.1. *An i-Riemann invariant $(1 \le i \le m)$ of the hyperbolic system (1.1) is a scalar-valued function $w(\mathbf{u})$ on $\mathcal{O}$, such that $\nabla w(\mathbf{u}) \cdot \mathbf{r}_i(\mathbf{u}) = 0$, $\forall u \in \mathcal{O}$, where $\mathbf{r}_i(\mathbf{u})$ is an eigenvector of the Jacobian matrix $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$ corresponding to the eigenvalue $\lambda_i(\mathbf{u})$.*

Riemann invariants are closely related to the Riemann problem, which is a Cauchy problem of the hyperbolic system (1.1) with the initial condition

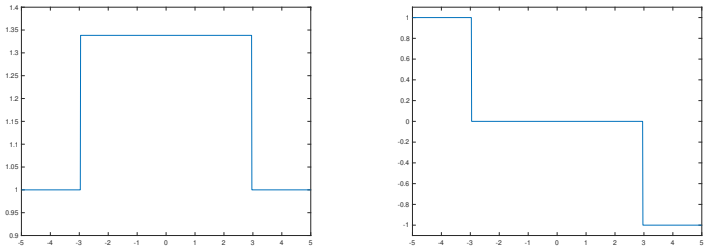
$$\mathbf{u}(x,0) = \begin{cases} \mathbf{u}_l, & x < 0 \\ \mathbf{u}_r, & x > 0 \end{cases}, \tag{2.1}$$

where $\mathbf{u}_l$ and $\mathbf{u}_r$ are constant states. It is well-known that the solution $\mathbf{u}(x,t)$ of the Riemann problem typically develops from the initial discontinuity at the origin into $m+1$ constant states in sector regions separated by the i-shock, contact or rarefaction

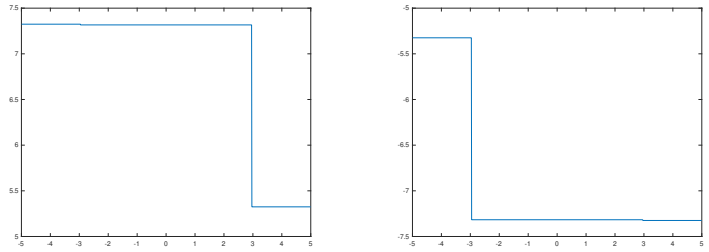
wave, for $i = 1, 2, \ldots, m$, which is a characterization of the fundamental behavior of solutions of hyperbolic systems involving discontinuities. An important property of Riemann invariants across waves is stated as follows [26]:

THEOREM 2.2. *The change of an i-Riemann invariant w of the hyperbolic system (1.1) across an i-shock wave is of third order in ϵ, i.e. $|w(\mathbf{u}_l) - w(\mathbf{u}_r)| = O(\epsilon^3)$, where $\mathbf{u}_l$ and $\mathbf{u}_r$ are the states on the left and right sides of the i-shock, respectively, and $\epsilon = |\lambda_i(\mathbf{u}_l) - \lambda_i(\mathbf{u}_r)|$ is a measure of the strength of the i-shock. In addition, the i-Riemann invariant is unchanged across an i-rarefaction or contact wave.*

Roughly speaking, the i-Riemann invariant is unchanged or almost unchanged across an i-wave, consult Figure 1, where h, hu are the conserved variables, and w_1, w_2 are the 1 and 2-Riemann invariants, respectively, in a Riemann problem of the shallow water equations (2.3).



(a) Water depth h (b) Discharge hu



(c) 1-Riemann invariant $w_1 = u - 2\sqrt{gh}$ (d) 2-Riemann invariant $w_2 = u + 2\sqrt{gh}$

Fig. 1: Conserved variables and Riemann invariants in a Riemann problem of the shallow water equations

The WENO interpolation/reconstruction procedure performs very well if there is only one discontinuity in the stencil. However, the results turn out to be less satisfactory when there are multiple shocks in the stencil. The property of Riemann invariants in Theorem 2.2 gives us a hint to perform the WENO procedure on the $1, 2$-Riemann invariants of hyperbolic systems when $m = 2$, as there is only one major discontinuity in each Riemann invariant in Riemann problems. We shall show in numerical section that such a treatment yields very satisfactory non-oscillatory results.

A direct extension of the above approach to hyperbolic systems with $m \geq 3$ is to perform the WENO procedure on m variables, each of which only admits one major

jump in stencils. An ideal choice is the coordinate system of Riemann invariants, which is defined as follows [7]:

DEFINITION 2.3. *The system* (1.1) *is endowed with a coordinate system of Riemann invariants if there exist* m *scalar-valued functions* $w_1(\mathbf{u}), w_2(\mathbf{u}), \ldots, w_m(\mathbf{u})$ *on* $\mathcal{O}$ *such that,*

$$\nabla w_i(\mathbf{u}) \cdot \mathbf{r}_j(\mathbf{u}) = \delta_{i,j}, \quad i, j = 1, 2, \ldots, m,$$

where δ *is the Kronecker delta,* $\mathbf{r}_j(\mathbf{u})$ *is an eigenvector of the Jacobian matrix* $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$ *corresponding to the eigenvalue* $\lambda_j(\mathbf{u}), 1 \le j \le m$*. The variables* $(w_1(\mathbf{u}), w_2(\mathbf{u}), \ldots, w_m(\mathbf{u}))$ *are called a coordinate system of Riemann invariants of* (1.1)*.*

To this end, we give some examples of hyperbolic systems of conservation laws endowed with a coordinate system of Riemann invariants.

EXAMPLE 2.1. *The linear hyperbolic system*

$$\mathbf{u}_t + \mathbf{A}\mathbf{u}_x = \mathbf{0}, \tag{2.2}$$

where $\mathbf{A} = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^{-1}$ *for some diagonal matrix* $\mathbf{\Lambda}$ *and eigenmatrix* $\mathbf{R}$*, has a coordinate system of Riemann invariants* $(w_1, w_2, \ldots, w_m)$ *with* $w_i(\mathbf{u}) = \mathbf{l}_i \mathbf{u}, 1 \le i \le m$*, where* $\mathbf{l}_i$ *is the* i*-th row of* $\mathbf{R}^{-1}$*.*

EXAMPLE 2.2. *The shallow water equations in one dimension*

$$\begin{pmatrix} h \\ hu \end{pmatrix}_t + \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{pmatrix}_x = \mathbf{0} \tag{2.3}$$

where h *is the water height,* u *is the velocity of the fluid, and* g *is the gravitational constant, is endowed with a coordinate system of Riemann invariants* $(w_1, w_2) = (u + 2\sqrt{gh}, u - 2\sqrt{gh})$*.*

The shallow water equations in two dimensions

$$\begin{pmatrix} h \\ hu \\ hv \end{pmatrix}_t + \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix}_x + \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix}_y = \mathbf{0}, \tag{2.4}$$

where u *and* v *are velocities of the fluid in* x *and* y *directions, respectively, has coordinate systems of Riemann invariants* $(w_1, w_2, w_3) = (u - 2\sqrt{gh}, v, u + 2\sqrt{gh})$ *and* $(w_1, w_2, w_3) = (v - 2\sqrt{gh}, u, v + 2\sqrt{gh})$ *in* x *and* y *directions, respectively, in the sense that the states of fluid are constant in the other direction (in this case, the system is of the form of one dimensional equations, which is known as the split multi-dimensional problem).*

EXAMPLE 2.3. *The hyperbolic system of electrophoresis of* m *components*

$$\partial_t u_i + \partial_x \left(\frac{c_i u_i}{\sum_{j=1}^{n} u_j} \right) = 0, \quad i = 1, 2, \ldots, m, \tag{2.5}$$

where $c_1 < c_2 < \cdots < c_m$ *are positive constants, is endowed with a coordinate system of Riemann invariants* $(w_1, w_2, \ldots, w_m)$*, where* $w_i \in (c_i, c_{i+1})$ *is the solution of the equation* $\sum_{j=1}^{m} \frac{u_j}{c_j - w} = 0$*, for* $i = 1, 2, \ldots, m-1$*, and* $w_m = \sum_{j=1}^{m} \frac{u_j}{c_j}$*.*

This system models the separation of ionized chemical compounds in solution driven by an electric filed, where c_i *and* u_i *denote the electrophoretic mobility and concentration of the* i*-th component, respectively, see [2] for more details about its physical backgrounds.*

EXAMPLE 2.4. *The hyperbolic system of planar electromagnetic waves in nonlinear isotropic dielectrics*

$$\begin{pmatrix} B_1 \\ B_2 \\ D_1 \\ D_2 \end{pmatrix}_t + \begin{pmatrix} -\frac{\Psi'(r)}{r} D_2 \\ \frac{\Psi'(r)}{r} D_1 \\ \frac{\Psi'(r)}{r} B_2 \\ -\frac{\Psi'(r)}{r} B_1 \end{pmatrix}_x = \mathbf{0}, \tag{2.6}$$

where $B = (B_1, B_2)^T$ is the magnetic induction, $D = (D_1, D_2)$ is the electric displacement, $\Psi(r)$ is the electromagnetic energy, and $r = \sqrt{B_1^2 + B_2^2 + D_1^2 + D_2^2}$, is endowed with a coordinate system of Riemann invariants (w_1, w_2, w_3, w_4).

If we define a, b, p, q by $pe^{ia} = \frac{1}{\sqrt{2}}(B_2 + D_1 - i(B_1 - D_2))$ and $qe^{ib} = \frac{1}{\sqrt{2}}(-B_2 + D_1 + i(B_1 + D_2))$, then $w_1 = a, w_2 = b$, and w_3, w_4 are the $1,2$-Riemann invariants of the smaller hyperbolic system

$$\begin{pmatrix} p \\ q \end{pmatrix}_t + \begin{pmatrix} \frac{\Psi'(r)}{r} p \\ -\frac{\Psi'(r)}{r} q \end{pmatrix}_x = \mathbf{0}, \quad r = \sqrt{p^2 + q^2}. \tag{2.7}$$

3. The algorithms. In this section, we overview the WENO-JS interpolation, and establish our algorithms in the framework of alternative formulation of finite difference WENO scheme in one and two space dimensions. We shall assume the grids are uniform and, for simplicity, only consider periodic boundaries.

3.1. Overview of the WENO-JS interpolation. The $(2r-1)$-th order WENO-JS interpolation for a scalar-valued grid function v is described as follows.

First, we define the small stencils $S_k = \{x_{j-r+k}, \ldots, x_{j-1+k}\}$ to calculate the $(r-1)$-th order polynomial interpolant $p^{(k)}(x)$ of v on I_j, for $k = 1, 2, \ldots, r$, and the big stencil $S_0 = \cup_{k=1}^r S_k = \{x_{j-r+1}, \ldots, x_{j+r-1}\}$ to calculate the $(2r-2)$-th order polynomial interpolant $p^{(0)}(x)$ of v on I_j, such that

$$p^{(k)}(x_{j-r+k+m-1}) = v_{j-r+k+m-1}, \quad m = 1, 2, \ldots, r,$$

for $k = 1, 2, \ldots, r$, and

$$p^{(0)}(x_{j-r+m}) = v_{j-r+m}, \quad m = 1, 2, \ldots, 2r-1,$$

so that we yield

(3.1)

$$v_{j+\frac{1}{2}}^{-(k)} = p^{(k)}(x_{j+\frac{1}{2}}) = \sum_{m=1}^{r} a_m^{(k)} v_{j-r+k+m-1} = v(x_{j+\frac{1}{2}}) + O(\Delta x^r), \quad k = 1, 2, \ldots, r,$$

and

$$v_{j+\frac{1}{2}}^{-(0)} = p^{(0)}(x_{j+\frac{1}{2}}) = \sum_{k=1}^{r} \gamma_k v_{j+\frac{1}{2}}^{-(k)} = v(x_{j+\frac{1}{2}}) + O(\Delta x^{2r-1}), \tag{3.2}$$

where $\{\gamma_k\}_{k=1}^r$ are the so-called optimal linear weights with $\gamma_k \geq 0$, for $k = 1, 2, \ldots, r$ [5] and $\sum_{k=1}^r \gamma_k = 1$, and $\{a_m^{(k)}\}_{m,k=1}^r$ are constant coefficients.

Then, we introduce the nonlinear weights $\{\omega_k\}_{k=1}^r$, which is designed in the principle that, in smooth regions w_k is close to γ_k to achieve optimal accuracy while,

if containing discontinuities, w_k is close to zero to minimize the contribution of the stencil containing discontinuities in WENO interpolation:

$$\omega_k = \frac{\tilde{\omega}_k}{\sum_{m=1}^{r} \tilde{\omega}_m}, \quad \tilde{\omega}_k = \frac{\gamma_k}{(\beta_k + \epsilon)^2}, \quad k = 1, 2, \ldots, r, \tag{3.3}$$

where ϵ is a small positive number, e.g. $\epsilon = 10^{-6}$, to avoid the case of linear weights being divided by zero, and $\{\beta_k\}_{k=1}^{r}$ are the smoothness indicators of the polynomial interpolant $p^{(k)}(x)$ on I_j:

$$\beta_k = \sum_{\ell=1}^{r} \Delta x^{2\ell-1} \int_{I_j} \left(\frac{d^\ell}{dx^\ell} p^{(k)}(x) \right)^2 dx. \tag{3.4}$$

Finally, the WENO-JS interpolation $v^-_{j+\frac{1}{2}}$ is calculated by

$$v^-_{j+\frac{1}{2}} = \sum_{k=1}^{r} \omega_k v^{-(k)}_{j+\frac{1}{2}}. \tag{3.5}$$

For instance, in the fifth order ($r = 3$) WENO-JS interpolation, we have

$$\begin{aligned}
v^{-(1)}_{j+\frac{1}{2}} &= \frac{3}{8} v_{j-2} - \frac{5}{4} v_{j-1} + \frac{15}{8} v_j, \\
v^{-(2)}_{j+\frac{1}{2}} &= -\frac{1}{8} v_{j-1} + \frac{3}{4} v_j + \frac{3}{8} v_{j+1}, \\
v^{-(3)}_{j+\frac{1}{2}} &= \frac{3}{8} v_j + \frac{3}{4} v_{j+1} - \frac{1}{8} v_{j+2},
\end{aligned}$$

and

$$\gamma_1 = \frac{1}{16}, \quad \gamma_2 = \frac{5}{8}, \quad \gamma_3 = \frac{5}{16},$$

and

$$\begin{aligned}
\beta_1 &= \frac{13}{12}(v_{j-2} - 2v_{j-1} + v_j)^2 + \frac{1}{4}(v_{j-2} - 4v_{j-1} + 3v_j)^2, \\
\beta_2 &= \frac{13}{12}(v_{j-1} - 2v_j + v_{j+1})^2 + \frac{1}{4}(v_{j-1} - v_{j+1})^2, \\
\beta_3 &= \frac{13}{12}(v_j - 2v_{j+1} + v_{j+2})^2 + \frac{1}{4}(3v_j - 4v_{j+1} + v_{j+2})^2.
\end{aligned}$$

For expressions of smoothness indicators in higher order WENO-JS interpolations, one can refer to [3].

3.2. The algorithm in one dimension. For the domain $[x_a, x_b]$, we take the uniform partition $x_a = x_0 < x_1 < \ldots < x_N = x_b$, and denote $\Delta x \equiv x_j - x_{j-1}$, $x_{j-\frac{1}{2}} = \frac{1}{2}(x_{j-1} + x_j)$, for $j = 1, 2, \ldots, N$. In the finite difference WENO scheme, we seek $\mathbf{u}_j$ to approximate $\mathbf{u}(x_j, t)$, and $\mathbf{u}^{\pm}_{j+\frac{1}{2}}$ to approximate the solution at $x_{j+\frac{1}{2}}$ from I_j and I_{j+1}, respectively. For the ease of writing, we shall use subscript indices exceeding the domain in the cyclic sense.

The semi-discrete $(2r-1)$-th order alternative formulation of finite difference WENO scheme for the hyperbolic system (1.1) in one dimensions is given by (1.2), in which we define

$$\hat{\mathbf{f}}_{j+\frac{1}{2}} = \mathbf{h}(\mathbf{u}^-_{j+\frac{1}{2}}, \mathbf{u}^+_{j+\frac{1}{2}}) + \sum_{m=1}^{r-1} a_{2m} \Delta x^{2m} \left(\frac{\partial^{2m}}{\partial x^{2m}} \mathbf{f} \right)_{j+\frac{1}{2}}, \tag{3.6}$$

where $\mathbf{h}(\cdot,\cdot)$ is the numerical flux based on exact or approximate Riemann solvers, e.g. the Godunov flux, the Lax-Friedrichs flux, or the HLLC-type fluxes, among others, and the coefficients $a_2 = -\frac{1}{24}, a_4 = \frac{7}{5760}, a_6 = -\frac{31}{967680}, a_8 = \frac{127}{154828800}, a_{10} = -\frac{73}{3503554560}, \ldots$, are obtained through Taylor expansion to approximate the spacial derivative of flux with high accuracy, see [24].

Following the practice in [18, 13], we calculate $\mathbf{u}^{\pm}_{j+\frac{1}{2}}$ in $\mathbf{h}(\mathbf{u}^{-}_{j+\frac{1}{2}}, \mathbf{u}^{+}_{j+\frac{1}{2}})$ by WENO interpolation, while use simple central difference to approximate the spatial derivatives of $\mathbf{f}$ in the remaining terms to save computational costs, as these terms contain at least Δx^2 in the coefficients, which is expected to contribute much less oscillations. To attain enough accuracy, we use the stencil $\{x_{j-r+1}, \ldots, x_j, \ldots, x_{j+r}\}$ in the central difference approximation for $\left(\frac{\partial^{2m}}{\partial x^{2m}}\mathbf{f}\right)_{j+\frac{1}{2}}$.

For instance, in the fifth order finite difference WENO, we use

$$\left(\frac{\partial^2}{\partial x^2}\mathbf{f}\right)_{j+\frac{1}{2}} = \frac{1}{\Delta x^2}\left(-\frac{5}{48}\mathbf{f}_{j-2} + \frac{13}{16}\mathbf{f}_{j-1} - \frac{17}{24}\mathbf{f}_j - \frac{17}{24}\mathbf{f}_{j+1} + \frac{13}{16}\mathbf{f}_{j+2} - \frac{5}{48}\mathbf{f}_{j+3}\right),$$

$$\left(\frac{\partial^4}{\partial x^4}\mathbf{f}\right)_{j+\frac{1}{2}} = \frac{1}{\Delta x^4}\left(\frac{1}{2}\mathbf{f}_{j-2} - \frac{3}{2}\mathbf{f}_{j-1} + \mathbf{f}_j + \mathbf{f}_{j+1} - \frac{3}{2}\mathbf{f}_{j+2} + \frac{1}{2}\mathbf{f}_{j+3}\right).$$

If the hyperbolic system (1.1) is endowed with a coordinate system of Riemann invariants $\mathbf{w}$ with the one-to-one algebraic relation $\mathbf{w} = \mathbf{w}(\mathbf{u})$ and $\mathbf{u} = \mathbf{u}(\mathbf{w})$ to the conserved variables $\mathbf{u}$, the $(2r-1)$-th order alternative formulation of finite difference WENO scheme based on the nodal values $\{\mathbf{u}^n_j\}_{j=1}^N$ at time level t^n is given as follows, where the superscript n is omitted for simplicity and computation is carried out for all $j = 1, 2, \ldots, N$:

1. Calculate the coordinate system of Riemann invariants $\mathbf{w}_j = \mathbf{w}(\mathbf{u}_j)$.
2. Perform the WENO interpolation introduced in Section 3.1 on $\{\mathbf{w}_j\}_{j=1}^N$ to obtain $\mathbf{w}^{-}_{j+\frac{1}{2}} = \text{weno}(\mathbf{w}_{j-r+1}, \ldots, \mathbf{w}_{j+r-1})$ and $\mathbf{w}^{+}_{j+\frac{1}{2}} = \text{weno}(\mathbf{w}_{j+r}, \ldots, \mathbf{w}_{j-r+2})$.
3. Transform the results back to the conserved variables by $\mathbf{u}^{\pm}_{j+\frac{1}{2}} = \mathbf{u}\left(\mathbf{w}^{\pm}_{j+\frac{1}{2}}\right)$.
4. Calculate the numerical fluxes $\hat{\mathbf{f}}_{j+\frac{1}{2}}$ to evolve the scheme (1.2) in time.

To this end, we would like to introduce the time-marching approach used the algorithm. For the ODE system,

$$\mathbf{u}_t = \mathbf{L}(\mathbf{u}), \tag{3.7}$$

which is obtained from the semi-discrete finite difference scheme, we adopt the 4-th order 5 stage strong stability preserving Runge-Kutta (SSPRK(4, 5)) method [27],

$$\begin{aligned}
\mathbf{u}^{(1)} &= \mathbf{u}^n + 0.39175222700392\Delta t\mathbf{L}(\mathbf{u}^n),\\
\mathbf{u}^{(2)} &= 0.44437049406734\mathbf{u}^n + 0.55562950593266\mathbf{u}^{(1)} + 0.36841059262959\Delta t\mathbf{L}(\mathbf{u}^{(1)}),\\
\mathbf{u}^{(3)} &= 0.62010185138540\mathbf{u}^n + 0.37989814861460\mathbf{u}^{(2)} + 0.25189177424738\Delta t\mathbf{L}(\mathbf{u}^{(2)}),\\
\mathbf{u}^{(4)} &= 0.17807995410773\mathbf{u}^n + 0.82192004589227\mathbf{u}^{(3)} + 0.54497475021237\Delta t\mathbf{L}(\mathbf{u}^{(3)}),\\
\mathbf{u}^{n+1} &= 0.00683325884039\mathbf{u}^n + 0.51723167208978\mathbf{u}^{(2)} + 0.12759831133288\mathbf{u}^{(3)} + 0.34833675773694\mathbf{u}^{(4)}\\
&\quad + 0.08460416338212\Delta t\mathbf{L}(\mathbf{u}^{(3)}) + 0.22600748319395\Delta t\mathbf{L}(\mathbf{u}^{(4)}),
\end{aligned}$$

where $\mathbf{u}^n$ and $\mathbf{u}^{n+1}$ are solutions at the time level t^n and t^{n+1}, respectively, and $\Delta t = t^{n+1} - t^n$. We refer to [8] and [9] for more details about the strong stability

preserving (SSP), also called the total variation diminishing (TVD), Runge-Kutta or multi-step time discretization approaches.

In the numerical section, we shall use WENO schemes with spatial accuracy higher than fourth order (the temporal accuracy), as in applications it is usually the spatial accuracy that restricts the resolution of simulations.

3.3. The algorithm in two dimensions. For the two dimensional domain $[x_a, x_b] \times [y_a, y_b]$, we take the uniform partition $x_a = x_0 < x_1 < \cdots < x_N = x_b$ and $y_a = y_0 < y_1 < \cdots < y_M = y_b$ in x and y directions, respectively, and denote by $\Delta x \equiv x_i - x_{i-1}$, $x_{i-\frac{1}{2}} = \frac{1}{2}(x_{i-1} + x_i)$ for $i = 1, 2, \ldots, N$, and $\Delta y \equiv y_j - y_{j-1}$, $y_{j-\frac{1}{2}} = \frac{1}{2}(y_{j-1} + y_j)$ for $j = 1, 2, \ldots, M$. We seek $\mathbf{u}_{i,j}$ to approximate $\mathbf{u}(x_i, y_j, t)$, and $\mathbf{u}^{\pm}_{i+\frac{1}{2},j}$ and $\mathbf{u}^{\pm}_{i,j+\frac{1}{2}}$ to approximate $\mathbf{u}(x_{i+\frac{1}{2}}, y_j, t)$ and $\mathbf{u}(x_i, y_{j+\frac{1}{2}}, t)$, respectively, from different sides, in the finite difference WENO schemes.

The semi-discrete $(2r-1)$-th order alternative formulation of finite difference WENO scheme for the hyperbolic system

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x + \mathbf{g}(\mathbf{u})_y = \mathbf{0}, \tag{3.8}$$

in two dimensions is formulated as

$$\frac{d\mathbf{u}_{i,j}}{dt} + \frac{1}{\Delta x}\left(\hat{\mathbf{f}}_{i+\frac{1}{2},j} - \hat{\mathbf{f}}_{i-\frac{1}{2},j}\right) + \frac{1}{\Delta y}\left(\hat{\mathbf{g}}_{i,j+\frac{1}{2}} - \hat{\mathbf{g}}_{i,j-\frac{1}{2}}\right) = \mathbf{0}, \tag{3.9}$$

for $i = 1, 2, \ldots, N, j = 1, 2, \ldots, M$, where the fluxes are defined the same way as in one dimensional case, thanks to the advantage of finite difference schemes.

If the x-split problem of (3.8) is endowed with a coordinate system of Riemann invariants $\mathbf{w}$ and the y-split problem of (3.8) is endowed with a coordinate system of Riemann invariants $\mathbf{v}$, the algorithm based on the nodal values $\{\mathbf{u}^n_{i,j}\}_{i=1,j=1}^{N,M}$ at time level t^n is given as follows, where the superscript n is omitted for brevity and computation is carried out for all $i = 1, 2, \ldots, N, j = 1, 2, \ldots, M$:

1. Calculate the coordinate systems of Riemann invariants $\mathbf{w}_{i,j} = \mathbf{w}(\mathbf{u}_{i,j})$ and $\mathbf{v}_{i,j} = \mathbf{v}(\mathbf{u}_{i,j})$.
2. Perform the WENO interpolation introduced in Section 3.1 on $\{\mathbf{w}_{i,j}\}_{i=1,j=1}^{N,M}$ and $\{\mathbf{v}_{i,j}\}_{i=1,j=1}^{N,M}$ to obtain $\mathbf{w}^-_{i+\frac{1}{2},j} = \text{weno}(\mathbf{w}_{i-r+1,j}, \ldots, \mathbf{w}_{i+r-1,j})$, $\mathbf{w}^+_{i+\frac{1}{2},j} = \text{weno}(\mathbf{w}_{i+r,j}, \ldots, \mathbf{w}_{i-r+2,j})$, and $\mathbf{v}^-_{i,j+\frac{1}{2}} = \text{weno}(\mathbf{v}_{i,j-r+1}, \ldots, \mathbf{v}_{i,j+r-1})$, $\mathbf{v}^+_{i,j+\frac{1}{2}} = \text{weno}(\mathbf{v}_{i,j+r}, \ldots, \mathbf{v}_{i,j-r+2})$.
3. Calculate $\mathbf{u}^{\pm}_{i+\frac{1}{2},j} = \mathbf{u}\left(\mathbf{w}^{\pm}_{i+\frac{1}{2},j}\right)$ and $\mathbf{u}^{\pm}_{i,j+\frac{1}{2}} = \mathbf{u}\left(\mathbf{v}^{\pm}_{i,j+\frac{1}{2}}\right)$.
4. Calculate the numerical fluxes $\hat{\mathbf{f}}_{i+\frac{1}{2},j}$ and $\hat{\mathbf{g}}_{i,j+\frac{1}{2}}$ to evolve the scheme (3.9) in time.

We adopt the same time marching approach in the algorithm as in the one space dimension.

4. Numerical tests. In this section, we study the accuracy, efficiency and essentially non-oscillatory performance of the algorithm established in the previous sections, and compare them with those of the component-wise and local characteristic decomposition based WENO methods. For convenience, the component-wise WENO, local characteristic decomposition based WENO and Riemann invariants based WENO methods shall be abbreviated to CW-WENO, LCD-WENO and RI-WENO, respectively. We adopt the Lax-Friedrichs flux as the lowest order term in the flux (3.6). The numerical tests are carried out for examples given in Section 2,

except for the first one, as the RI-WENO and LCD-WENO are exactly the same for linear hyperbolic systems.

EXAMPLE 4.1. *(Accuracy and efficiency)*

In this example, we compare the accuracy and efficiency of RI-WENO with those of the CW-WENO and LCD-WENO for the one dimensional shallow water equations (2.3).

It is easy to verify that, if $v(x,t)$ *is a classic solution of the inviscid Burgers' equation* $v_t + \left(\frac{v^2}{2}\right)_x = 0$, *then* $h(x,t) = \frac{4}{9}v^2(x,t)$ *and* $u(x,t) = \frac{2}{3}v(x,t)$ *are solutions of the shallow water equations with the gravitational constant* $g = \frac{1}{4}$, *thus we let* $v(x,0) = \frac{1}{2}\sin(x) + 1$ *to determine the corresponding initial conditions of* h *and* u. *We set the the domain* $\Omega = [0, 2\pi]$ *and enforce the periodic boundary condition in the tests. The CFL conditions are taken as* $\Delta t = \frac{1}{10\lambda_{\max}}\Delta x^{\frac{2r-1}{4}}$ *in accuracy tests and* $\Delta t = \frac{1}{10\lambda_{\max}}\Delta x$ *in efficiency tests, where* $\lambda_{\max} = || \left(|u| + \sqrt{gh}\right) ||_\infty$, *and the terminal time is* $T = 0.1$.

The errors and orders of convergence of CW-WENO, RI-WENO and LCD-WENO for h *are given in Table 1, from which we can clearly observe that RI-WENO has the same orders of convergence as those of CW-WENO.*

Moreover, we compare the CPU times of CW-WENO, RI-WENO and LCD-WENO on different grids for different orders. The code is run on Oscar[1] with 1 *core and* 8*GB memory, and we count the CPU times by taking the average of* 1000 *trials of the complete computation. The results are given in Table 2, from which we can see that RI-WENO has roughly the same efficiency as CW-WENO while reduces considerable computational costs from LCD-WENO.*

	method	CW-WENO		RI-WENO		LCD-WENO	
r	N	L^1 error	order	L^1 error	order	L^1 error	order
3	20	5.08E-04	-	1.07E-04	-	1.29E-03	-
	40	1.53E-05	5.05	3.12E-06	5.10	9.23E-05	3.80
	80	4.12E-07	5.22	9.18E-08	5.08	6.03E-06	3.94
	160	1.16E-08	5.16	2.79E-09	5.04	3.64E-07	4.05
	200	3.71E-09	5.09	9.28E-10	4.93	1.23E-07	4.87
4	10	4.09E-03	-	8.43E-04	-	4.53E-03	-
	20	7.82E-05	5.71	6.90E-06	6.93	2.88E-04	3.98
	40	7.66E-07	6.67	6.42E-08	6.75	1.78E-05	4.01
	60	5.81E-08	6.36	5.61E-09	6.01	3.63E-06	3.92
5	10	1.75E-03	-	3.42E-04	-	2.12E-03	-
	20	8.11E-06	7.76	8.91E-07	8.58	3.89E-05	5.77
	30	1.87E-07	9.30	2.04E-08	9.32	3.26E-06	6.12
	40	1.41E-08	9.00	1.28E-09	9.62	3.65E-07	7.61
6	12	2.11E-04	-	3.86E-05	-	2.84E-04	-
	20	1.93E-06	9.20	2.68E-07	9.73	9.25E-06	6.70
	30	1.89E-08	11.40	2.92E-09	11.15	4.76E-07	7.32
	40	9.34E-10	10.46	1.09E-10	11.41	3.81E-08	8.78

Table 1: Accuracy of h of different WENO methods in Example 4.1

EXAMPLE 4.2. *(Shallow water equations in one dimension)*

In this test, we compare the essentially non-oscillatory performance of RI-WENO with that of CW-WENO and LCD-WENO for the shallow water equations (2.3) *in*

	method	CW-WENO	RI-WENO	LCD-WENO
r	N	CPU time (s)	CPU time (s)	CPU time (s)
3	50	1.58E-03	1.74E-03	3.55E-03
	100	6.22E-03	6.52E-03	1.44E-02
	150	9.94E-03	1.07E-02	2.77E-02
	200	1.72E-02	1.84E-02	4.87E-02
4	50	2.78E-03	2.97E-03	5.27E-03
	100	1.08E-02	1.13E-02	2.11E-02
	150	2.03E-02	2.10E-02	4.35E-02
	200	3.35E-02	3.69E-02	6.76E-02
5	50	3.78E-03	3.95E-03	6.50E-03
	100	1.47E-02	1.52E-02	2.60E-02
	150	2.88E-02	2.96E-02	5.42E-02
	200	4.65E-02	5.18E-02	8.38E-02
6	50	4.84E-03	5.01E-03	7.94E-03
	100	1.90E-02	1.95E-02	3.18E-02
	150	3.81E-02	3.88E-02	6.71E-02
	200	6.57E-02	6.76E-02	1.03E-01

Table 2: CPU times of different WENO methods in Example 4.1

one dimension.

We first solve a Riemann problem with $g = 10$ and the initial condition

$$h(x,0) = \begin{cases} 0.125, & x < 0 \\ 1.000, & x > 0 \end{cases}, \quad u(x,0) = 0,$$

on the domain $\Omega = [-5, 5]$ with the partition $N = 200$. The plots of h of different methods at $T = 1$ are compared in Figure 2, where the reference solution are given by the exact Riemann solver.

We then solve a periodic boundary problem with $g = 1$ and the initial condition

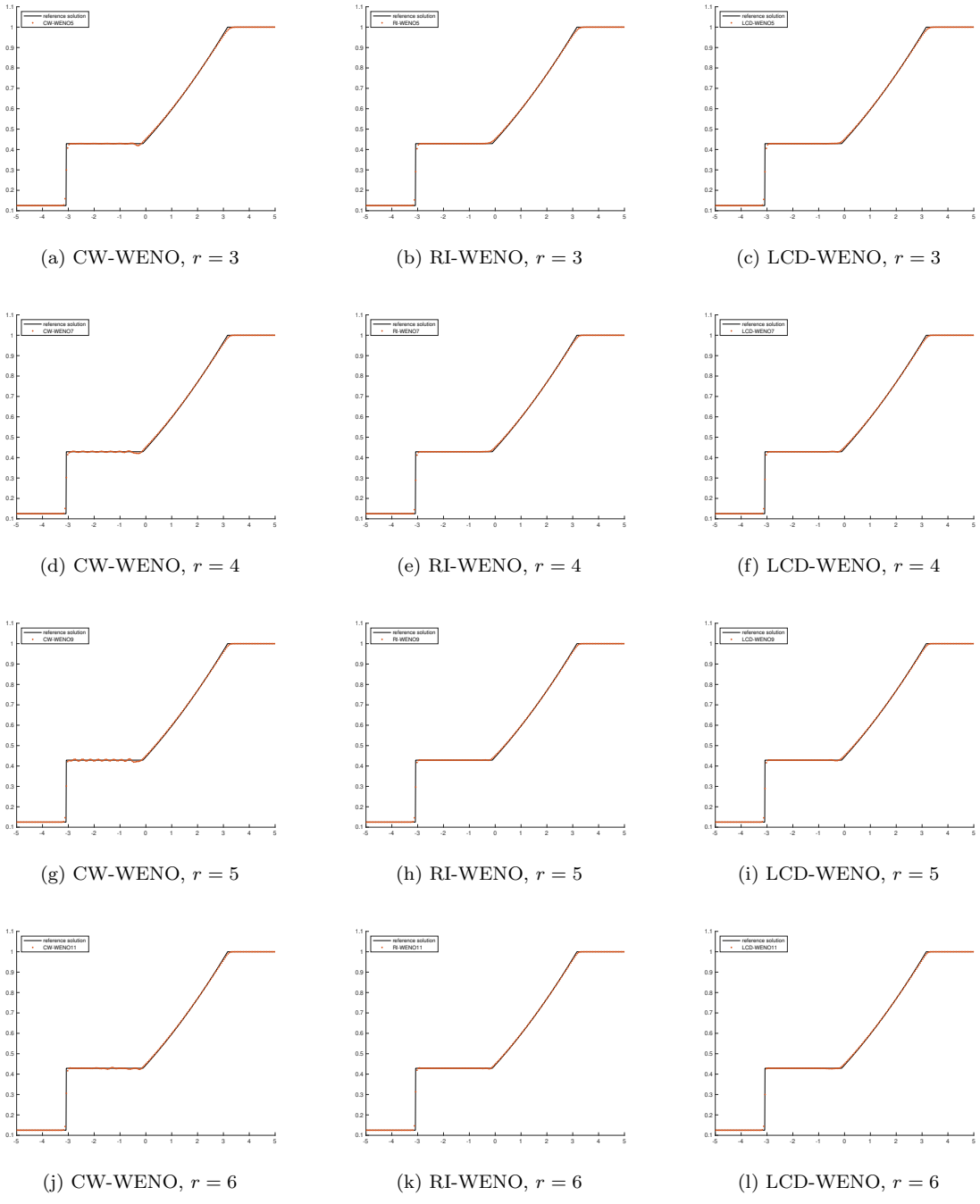
$$h(x,0) = \begin{cases} 2.0, & 0 < x < 10 \\ 1.5, & 10 < x < 20 \end{cases}, \quad u(x,0) = 0,$$

on the domain $\Omega = [0, 20]$ with the partition $N = 200$. The plots of h of different methods at $T = 20$ are compared in Figure 3, where the reference solution is obtained from the fifth order LCD-WENO on a grid containing 10000 cells.

By comparison, we observe the essentially non-oscillatory effect of RI-WENO is much better than CW-WENO, and similar to LCD-WENO.

EXAMPLE 4.3. *(Shallow water equations in two dimensions)*

In this test, we compare the essentially non-oscillatory performance of RI-WENO with that of CW-WENO and LCD-WENO for the shallow water equations (2.4) in two dimensions.



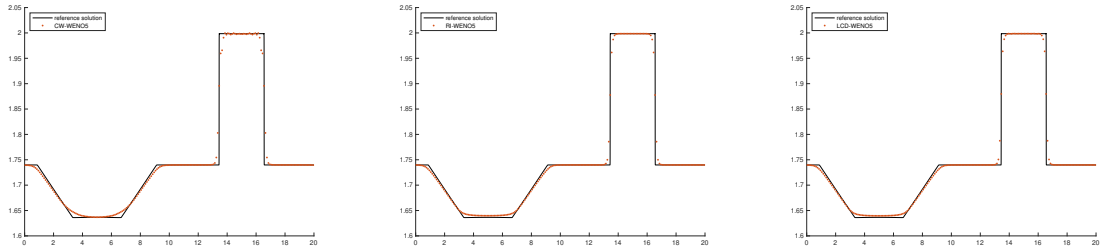
(a) CW-WENO, $r = 3$ (b) RI-WENO, $r = 3$ (c) LCD-WENO, $r = 3$

(d) CW-WENO, $r = 4$ (e) RI-WENO, $r = 4$ (f) LCD-WENO, $r = 4$

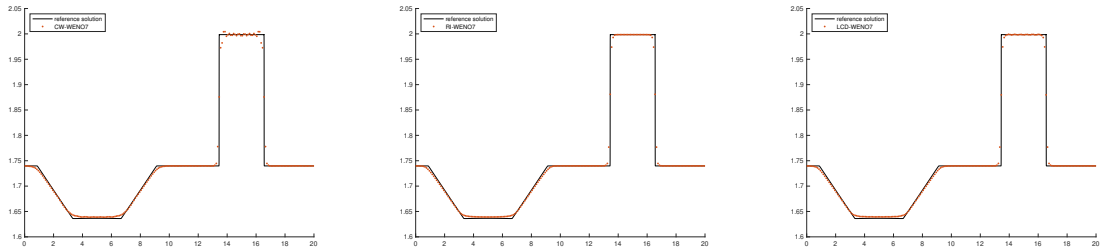
(g) CW-WENO, $r = 5$ (h) RI-WENO, $r = 5$ (i) LCD-WENO, $r = 5$

(j) CW-WENO, $r = 6$ (k) RI-WENO, $r = 6$ (l) LCD-WENO, $r = 6$

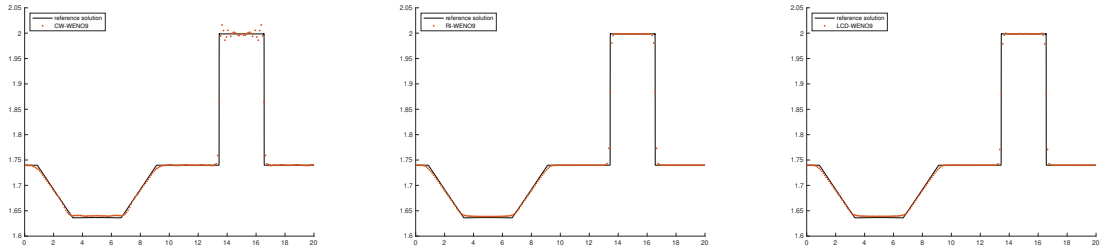
Fig. 2: Solution h of different WENO methods for the Riemann problem in Example 4.2.



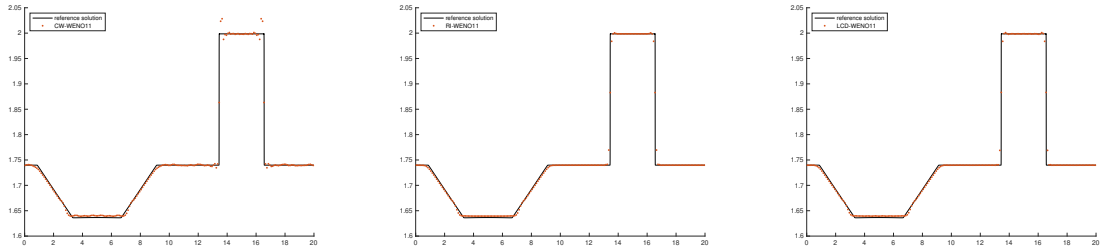
(a) CW-WENO, $r = 3$ (b) RI-WENO, $r = 3$ (c) LCD-WENO, $r = 3$



(d) CW-WENO, $r = 4$ (e) RI-WENO, $r = 4$ (f) LCD-WENO, $r = 4$



(g) CW-WENO, $r = 5$ (h) RI-WENO, $r = 5$ (i) LCD-WENO, $r = 5$



(j) CW-WENO, $r = 6$ (k) RI-WENO, $r = 6$ (l) LCD-WENO, $r = 6$

Fig. 3: Solution h of different WENO methods for the periodic boundary problem in Example 4.2.

We solve a periodic boundary problem with $g = 1$ and the initial condition

$$h(x,y,0) = \begin{cases} 2.5, & 0 < x < 10, 0 < y < 10 \\ 2.0, & 0 < x < 10, 10 < y < 20 \\ 0.5, & 10 < x < 20, 0 < y < 10 \\ 1.5, & 10 < x < 20, 10 < y < 20 \end{cases}, \quad u(x,y,0) = v(x,y,0) = 0,$$

on the domain $\Omega = [0,20]^2$ with $N = M = 200$.

The contours of h of different methods at $T = 5$ are shown in Figure 4, from which we can observe oscillations in the fourth quadrant in CW-WENO are eliminated by RI-WENO and LCD-WENO. Moreover, we plot the cut of h along $y = 10$ for different methods, and compare them with the reference solution obtained from the fifth order LCD-WENO on a 1000×1000 grid in Figure 5, from which we can see the non-oscillatory fashion of RI-WENO.

EXAMPLE 4.4. *(Equations of electrophoresis)*

In this test, we compare the essentially non-oscillatory performance of RI-WENO with that of CW-WENO and LCD-WENO for the electrophoresis equations (2.5).

We solve the three-component periodic boundary problem with the electrophoretic mobilities $c_1 = 2, c_2 = 4, c_3 = 5$, and the initial condition

$$u_1(x,0) = \begin{cases} 1, & 0 < x < \frac{\pi}{2} \\ 0.01, & \frac{\pi}{2} < x < 2\pi \end{cases}, u_2(x,0) = \begin{cases} 0.01, & 0 < x < \frac{3\pi}{2} \\ 1, & \frac{3\pi}{2} < x < 2\pi \end{cases}, u_3(x,0) = 1,$$

on the domain $\Omega = [0, 2\pi]$ with $N = 200$.

The plots of u_1 of different methods at $T = 0.5$ are compared in Figure 6, where the reference solution is obtained from the fifth order LCD-WENO on a grid containing 10000 *cells. The results of RI-WENO apparently have much less oscillation compared with those of CW-WENO and similar fashion with LCD-WENO.*

EXAMPLE 4.5. *(Equations of planar electromagnetic wave)*

In this test, we compare the essentially non-oscillatory performance of RI-WENO with that of CW-WENO and LCD-WENO for the planar electromagnetic wave equations (2.6). *One can check that, if the electromagnetic energy satisfies $\frac{\Psi'(r)}{r} = r^\alpha$ for some $\alpha > 0$, the $1,2$-Riemann invariants of the smaller hyperbolic system in Example 2.4 have the expressions $w_3(p,q) = p - qG^{-1}(\log \frac{1}{q})$ and $w_4(p,q) = p + qG^{-1}(\log \frac{1}{q})$, where $G(\cdot)$ is defined in the Appendix B.*

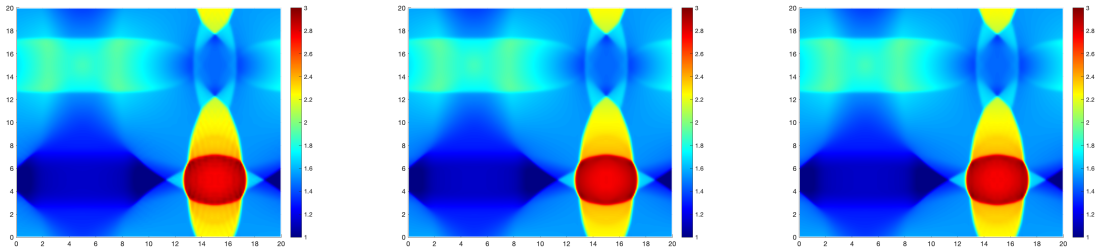
We solve the periodic boundary problem with $\alpha = 2$ and the initial condition

$$B_1(x,0) = \begin{cases} 1, & 0 < x < 2 \\ 0, & 2 < x < 4 \end{cases}, \quad B_2(x,0) = D_1(x,0) = D_2(x,0) = 1,$$

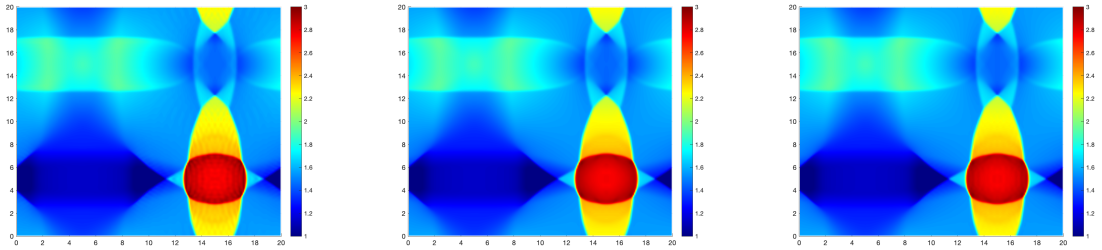
on the domain $\Omega = [0,4]$ with $N = 400$.

The plots of D_1 of different methods at $T = 0.3$ are compared in Figure 7, where the reference solution is obtained from the fifth order LCD-WENO on a grid containing 10000 *cells. From the comparison, we can see that RI-WENO has excellent essentially non-oscillatory performance.*

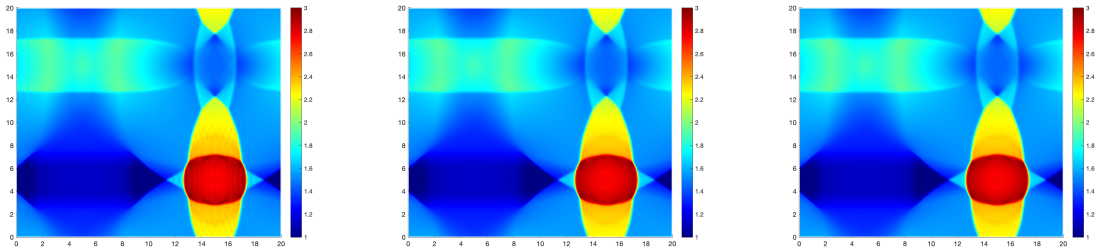
5. Concluding remarks. In this work, we establish a local characteristic decomposition free WENO method for hyperbolic system of conservation laws endowed with a coordinate system of Riemann invariants. We apply the WENO procedure to



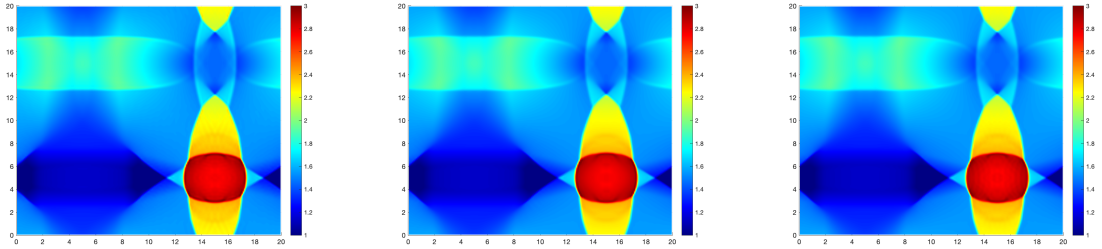
(a) CW-WENO, $r = 3$ (b) RI-WENO, $r = 3$ (c) LCD-WENO, $r = 3$



(d) CW-WENO, $r = 4$ (e) RI-WENO, $r = 4$ (f) LCD-WENO, $r = 4$

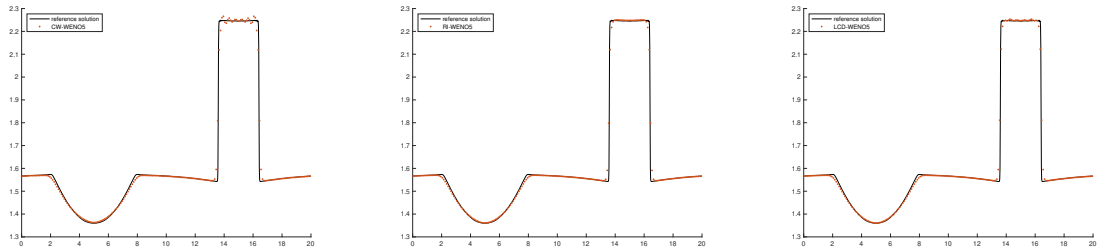


(g) CW-WENO, $r = 5$ (h) RI-WENO, $r = 5$ (i) LCD-WENO, $r = 5$

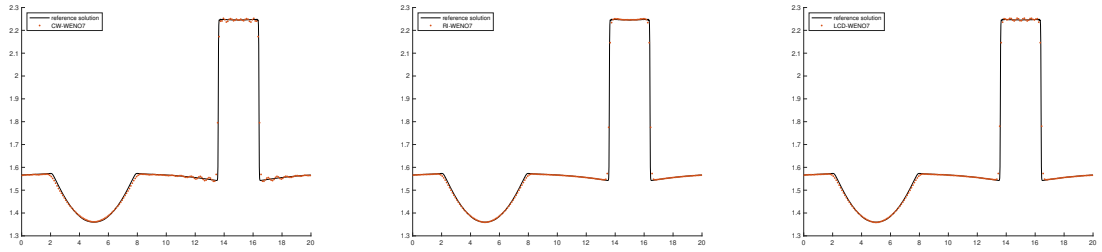


(j) CW-WENO, $r = 6$ (k) RI-WENO, $r = 6$ (l) LCD-WENO, $r = 6$

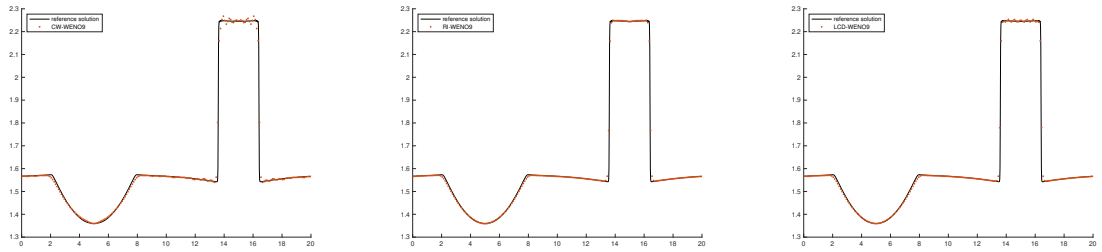
Fig. 4: Contours of h of difference WENO methods in Example 4.3.



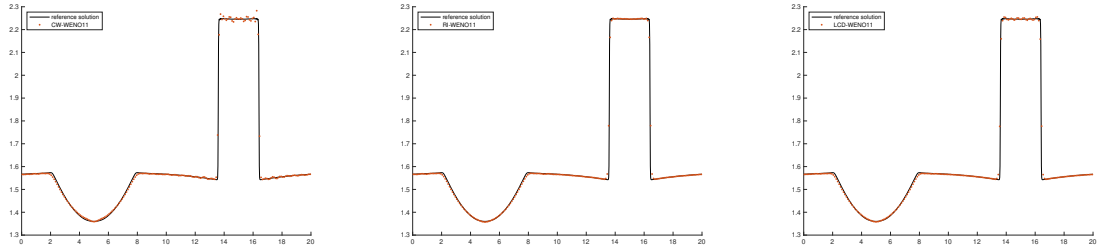
(a) CW-WENO, $r = 3$ (b) RI-WENO, $r = 3$ (c) LCD-WENO, $r = 3$



(d) CW-WENO, $r = 4$ (e) RI-WENO, $r = 4$ (f) LCD-WENO, $r = 4$

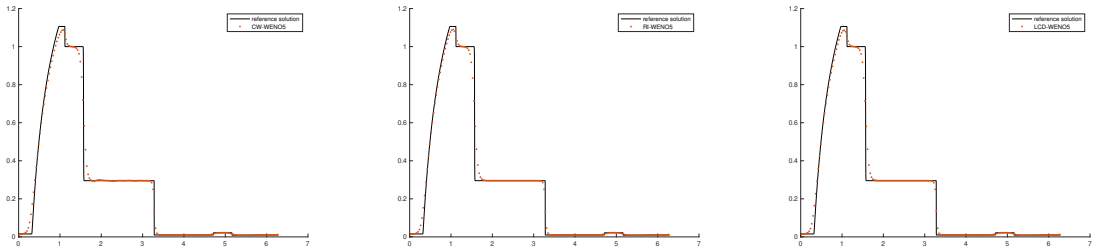


(g) CW-WENO, $r = 5$ (h) RI-WENO, $r = 5$ (i) LCD-WENO, $r = 5$

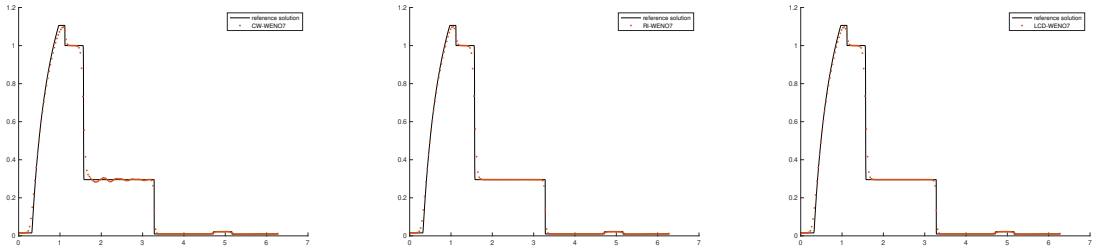


(j) CW-WENO, $r = 6$ (k) RI-WENO, $r = 6$ (l) LCD-WENO, $r = 6$

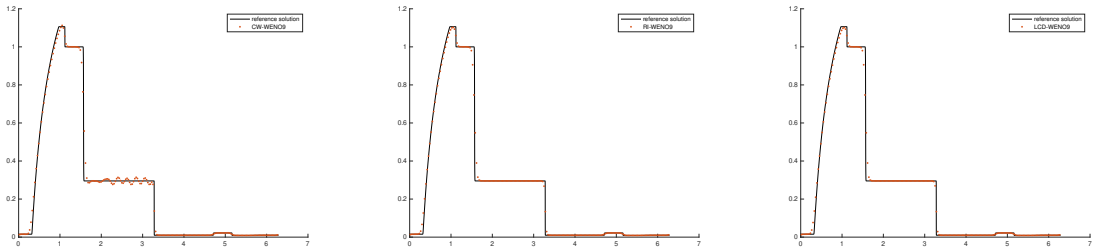
Fig. 5: Cut of h along $y = 10$ of difference WENO methods in Example 4.3.



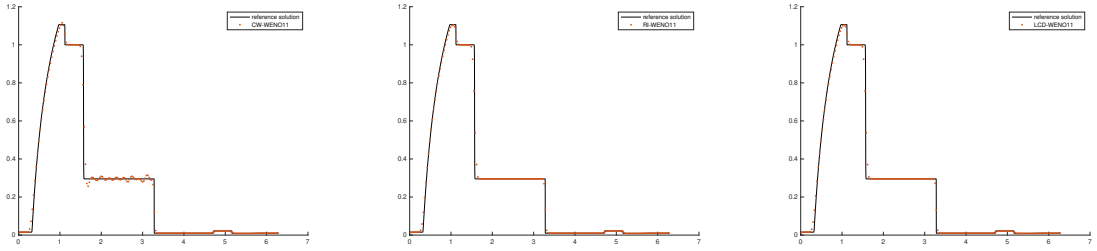
(a) CW-WENO, $r = 3$ (b) RI-WENO, $r = 3$ (c) LCD-WENO, $r = 3$



(d) CW-WENO, $r = 4$ (e) RI-WENO, $r = 4$ (f) LCD-WENO, $r = 4$

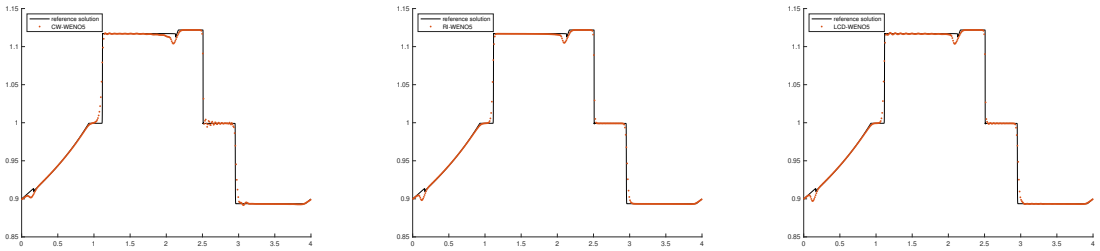


(g) CW-WENO, $r = 5$ (h) RI-WENO, $r = 5$ (i) LCD-WENO, $r = 5$

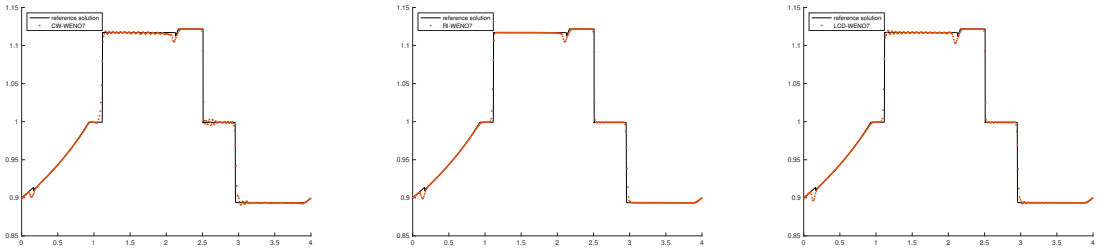


(j) CW-WENO, $r = 6$ (k) RI-WENO, $r = 6$ (l) LCD-WENO, $r = 6$

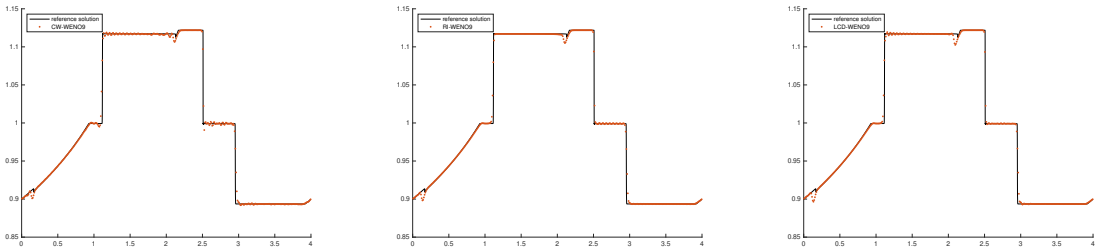
Fig. 6: Solution u_1 of different WENO methods in Example 4.4.



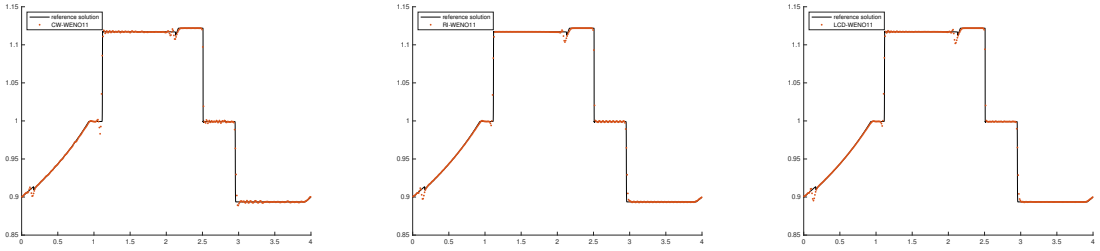
(a) CW-WENO, $r = 3$ (b) RI-WENO, $r = 3$ (c) LCD-WENO, $r = 3$



(d) CW-WENO, $r = 4$ (e) RI-WENO, $r = 4$ (f) LCD-WENO, $r = 4$



(g) CW-WENO, $r = 5$ (h) RI-WENO, $r = 5$ (i) LCD-WENO, $r = 5$



(j) CW-WENO, $r = 6$ (k) RI-WENO, $r = 6$ (l) LCD-WENO, $r = 6$

Fig. 7: Solution D_1 of different WENO methods in Example 4.5.

the coordinate system of Riemann invariants instead of the local characteristic fields of the hyperbolic system, thereby the efficiency is improved significantly. Due to the nonlinear algebraic relation of Riemann invariants and conserved variables/fluxes, we have to adopt the interpolation based alternative formulation of finite difference WENO method. Numerical tests show that the Riemann invariants based WENO method has optimal order of convergence and roughly the same efficiency as that of the components-wise WENO, but its essentially non-oscillatory fashion is similar to that of local characteristic decomposition based WENO.

Appendix A. A comparison of operations in LCD-WENO and RI-WENO for one dimensional shallow water equations. We analyze and compare the floating point operations in the local characteristic decomposition based WENO (LCD-WENO) and Riemann invariants based WENO (RI-WENO) algorithms for one dimensional shallow water equations in Table 3. From comparison, it is

steps	LCD-WENO	RI-WENO
1	$\mathbf{u}=\frac{1}{2}(\mathbf{u}_j+\mathbf{u}_{j+1})$, or Roe's average.	None
2	$\mathbf{R}(\mathbf{u})=\begin{bmatrix}1 & 1\\ u-\sqrt{gh} & u+\sqrt{gh}\end{bmatrix}$, $\mathbf{R}^{-1}(\mathbf{u})=\begin{bmatrix}\frac{1}{2}+\frac{u}{2\sqrt{gh}} & -\frac{1}{2\sqrt{gh}}\\ \frac{1}{2}-\frac{u}{2\sqrt{gh}} & \frac{1}{2\sqrt{gh}}\end{bmatrix}$.	$w_1=u+2\sqrt{gh}$, $w_2=u-2\sqrt{gh}$.
3	$\mathbf{v}_i=\mathbf{R}^{-1}\mathbf{u}_i,\ i=j-r+1,\dots,j+r.$	None
4	$\mathbf{v}^-_{j+\frac{1}{2}}=\text{weno}(\mathbf{v}_{j-r+1},\dots,\mathbf{v}_{j+r-1})$, $\mathbf{v}^+_{j+\frac{1}{2}}=\text{weno}(\mathbf{v}_{j+r},\dots,\mathbf{v}_{j-r+2})$	$\mathbf{w}^-_{j+\frac{1}{2}}=\text{weno}(\mathbf{w}_{j-r+1},\dots,\mathbf{w}_{j+r-1})$, $\mathbf{w}^+_{j+\frac{1}{2}}=\text{weno}(\mathbf{w}_{j+r},\dots,\mathbf{w}_{j-r+2})$
5	$\mathbf{u}^\pm_{j+\frac{1}{2}}=\mathbf{R}\mathbf{v}^\pm_{j+\frac{1}{2}}$	$\mathbf{u}^\pm_{j+\frac{1}{2}}=\mathbf{u}\left(\mathbf{w}^\pm_{j+\frac{1}{2}}\right)$
6	$\hat{\mathbf{f}}(\mathbf{u}^-_{j+\frac{1}{2}},\mathbf{u}^+_{j+\frac{1}{2}},\cdots)$	$\hat{\mathbf{f}}(\mathbf{u}^-_{j+\frac{1}{2}},\mathbf{u}^+_{j+\frac{1}{2}},\cdots)$

Table 3: Comparison of operations in LCD-WENO and RI-WENO algorithms for one dimensional shallow water equations

clear that, RI-WENO exempts the computations at steps 1 and 3, saves computational costs at step 2, and has exactly the same costs at steps 4, 5 and 6. (At step 5, both algorithms use 4 multiplications and two additions, due to the relation $h=c\,(w_1-w_2)^2\,, u=\frac{1}{2}(w_1+w_2), hu=h*u$, where $c=\frac{1}{16g}$.)

Appendix B. The definition of $G(\cdot)$ and computation of $G^{-1}(\cdot)$ in Example 4.5 . Let

$$g(u)=\frac{1}{(1+\frac{2+\alpha}{2\alpha})u+\frac{2+\alpha}{2\alpha}u^{-1}-\sqrt{\left(\frac{2+\alpha}{2\alpha}\right)^2u^2+\left(\frac{2+\alpha}{2\alpha}\right)^2u^{-2}+\frac{8+8\alpha-2\alpha^2}{4\alpha^2}}},\quad u>0,$$

then

$$
\begin{aligned}
G(u) &= \int_1^u g(y)dy \\
&= \frac{1}{16(1+\alpha)}\left(-\alpha\log 16 + (8+4\alpha)\log u + 4\alpha\log(1+u^2) + 2\alpha\log\left(\frac{\alpha-\alpha u^2+t}{-\alpha+\alpha u^2+t}\right)\right. \\
&+ (\alpha+2)\log\left(\frac{-\alpha^2-4\alpha-4+(\alpha^2-4\alpha-4)u^2+(2+\alpha)t}{\alpha^2-4\alpha-4+(-\alpha^2-4\alpha-4)u^2+(2+\alpha)t}\right) \\
&\left.+(\alpha+2)\log\left(\frac{-\alpha^2+4\alpha+4+(\alpha^2+4\alpha+4)u^2+(2+\alpha)t}{\alpha^2+4\alpha+4+(-\alpha^2+4\alpha+4)u^2+(2+\alpha)t}\right)\right),
\end{aligned}
$$

where $t = \sqrt{\alpha^2(u^2-1)^2 + (4\alpha+4)(u^2+1)^2}$.

Note that $G(u)$ is a log-like monotone increasing concave function with $\left(\frac{2+\alpha}{2+2\alpha}\right)u^{-1} < g(u) < u^{-1}$ for $u \in (0,\infty)$, and $\lim_{u\to 0^+}\frac{g(u)}{\left(\frac{2+\alpha}{2+2\alpha}\right)u^{-1}} = 1$, $\lim_{u\to\infty}\frac{g(u)}{u^{-1}} = 1$, thus one can compute $G^{-1}(\log\frac{1}{q})$ by solving u from the equation $G(u) + \log q = 0$ based on the Newton iteration.

REFERENCES

[1] *Ocean State Center for Advanced Resources*. https://docs.ccv.brown.edu/oscar/.

[2] V. G. Babskii, M. Y. Zhukov, and V. I. Yudovich, *Mathematical theory of electrophoresis*, Consultants Bureau, New York, 1989.

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

[4] R. Borges, M. Carmona, B. Costa, and W. S. Don, *An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws*, Journal of Computational Physics, 227 (2008), pp. 3191–3211.

[5] E. Carlini, R. Ferretti, and G. Russo, *A weighted essentially nonoscillatory, large time-step scheme for Hamilton–Jacobi equations*, SIAM Journal on Scientific Computing, 27 (2005), pp. 1071–1091.

[6] M. Castro, B. Costa, and W. S. Don, *High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws*, Journal of Computational Physics, 230 (2011), pp. 1766–1792.

[7] C. M. Dafermos, *Hyperbolic conservation laws in continuum physics*, Series: Grundlehren der mathematischen Wissenschaften 325, Springer-Verlag, Berlin, 2005.

[8] S. Gottlieb, D. Ketcheson, and C.-W. Shu, *High order strong stability preserving time discretizations*, Journal of Scientific Computing, 38 (2009), pp. 251–289.

[9] S. Gottlieb, D. Ketcheson, and C.-W. Shu, *Strong stability preserving Runge-Kutta and multistep time discretizations*, World Scientific, 2011.

[10] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, *Uniformly high order essentially non-oscillatory schemes, III*, Journal of Computational Physics, 71 (1987), pp. 231–303.

[11] A. K. Henrick, T. D. Aslam, and J. M. Powers, *Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points*, Journal of Computational Physics, 207 (2005), pp. 542–567.

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

[13] Y. Jiang, C.-W. Shu, and M. Zhang, *An alternative formulation of finite difference weighted ENO schemes with Lax–Wendroff time discretization for conservation laws*, SIAM Journal on Scientific Computing, 35 (2013), pp. A1137–A1160.

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

[15] X.-D. Liu and E. Tadmor, *Third order nonoscillatory central scheme for hyperbolic conservation laws*, Numerische Mathematik, 79 (1998), pp. 397–425.

[16] H. Nessyahu and E. Tadmor, *Non-oscillatory central differencing for hyperbolic conservation laws*, Journal of Computational Physics, 87 (1990), pp. 408–463.
[17] J. Qiu and C.-W. Shu, *On the construction, comparison, and local characteristic decomposition for high-order central WENO schemes*, Journal of Computational Physics, 183 (2002), pp. 187–209.
[18] J. Qiu and C.-W. Shu, *Finite difference WENO schemes with Lax–Wendroff-type time discretizations*, SIAM Journal on Scientific Computing, 24 (2003), pp. 2185–2198.
[19] J. Qiu and C.-W. Shu, *Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method: One-dimensional case*, Journal of Computational Physics, 193 (2004), pp. 115–135.
[20] P. Roe, *Approximate Riemann solvers, parameter vectors, and difference schemes*, Journal of Computational Physics, 27 (1978), pp. 1–31.
[21] J. Shi, C. Hu, and C.-W. Shu, *A technique of treating negative weights in WENO schemes*, Journal of Computational Physics, 175 (2002), pp. 108–127.
[22] C.-W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, in *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, B. Cockburn, C. Johnson, C.-W. Shu and E. Tadmor (Editor: A. Quarteroni), Lecture Notes in Mathematics, volume 1697, Springer, Berlin, 1998, pp.325-432.
[23] C.-W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes*, Acta Numerica, 29 (2020), pp. 701–762.
[24] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, Journal of Computational Physics, 77 (1988), pp. 439–471.
[25] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes, II*, Journal of Computational Physics, 83 (1989), pp. 32–78.
[26] J. Smoller, *Shock waves and reactiondiffusion equations*, Springer-Verlag, New York, 1983.
[27] R. J. Spiteri and S. J. Ruuth, *A new class of optimal high-order strong-stability-preserving time discretization methods*, SIAM Journal on Numerical Analysis, 40 (2002), pp. 469–491.
[28] F. Zheng, C.-W. Shu, and J. Qiu, *A high order conservative finite difference scheme for compressible two-medium flows*, Journal of Computational Physics, 445 (2021), p. 110597.
[29] J. Zhu and C.-W. Shu, *A new type of multi-resolution WENO schemes with increasingly higher order of accuracy*, Journal of Computational Physics, 375 (2018), pp. 659–683.