Journal of Computational Physics 498 (2024) 112669

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Provably convergent Newton–Raphson methods for recovering primitive variables with applications to physical-constraint-preserving Hermite WENO schemes for relativistic hydrodynamics ☆

Chaoyi Cai [a], Jianxian Qiu [b], Kailiang Wu [c,*]

[a] School of Mathematical Sciences, Xiamen University, Xiamen, Fujian 361005, PR China
[b] School of Mathematical Sciences and Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computing, Xiamen University, Xiamen, Fujian 361005, PR China
[c] Department of Mathematics and SUSTech International Center for Mathematics, Southern University of Science and Technology, and National Center for Applied Mathematics Shenzhen (NCAMS), Shenzhen 518055, PR China

ARTICLE INFO

Keywords:
Relativistic hydrodynamics
Physical-constraint-preserving
Newton–Raphson method
Hermite WENO
High-order accuracy
Finite volume scheme

ABSTRACT

The relativistic hydrodynamics (RHD) equations have three crucial intrinsic physical constraints on the primitive variables: positivity of pressure and density, and subluminal fluid velocity. However, numerical simulations can violate these constraints, leading to nonphysical results or even simulation failure. Designing genuinely physical-constraint-preserving (PCP) schemes is very difficult, as the primitive variables cannot be explicitly reformulated using conservative variables due to relativistic effects. In this paper, we propose three efficient Newton–Raphson (NR) methods for robustly recovering primitive variables from conservative variables. Importantly, we rigorously prove that these NR methods are always convergent and PCP, meaning they preserve the physical constraints throughout the NR iterations. The discovery of these robust NR methods and their PCP convergence analyses are highly nontrivial and technical. Our NR methods are versatile and can be seamlessly incorporated into any RHD schemes that require the recovery of primitive variables. As an application, we apply them to design PCP finite volume Hermite weighted essentially non-oscillatory (HWENO) schemes for solving the RHD equations. Our PCP HWENO schemes incorporate high-order HWENO reconstruction, a PCP limiter, and strong-stability-preserving time discretization. We rigorously prove the PCP property of the fully discrete schemes using convex decomposition techniques. Moreover, we suggest the characteristic decomposition with rescaled eigenvectors and scale-invariant nonlinear weights to enhance the performance of the HWENO schemes in simulating large-scale RHD problems. Several demanding numerical tests are conducted to demonstrate the robustness, accuracy, and high resolution of the proposed PCP HWENO schemes and to validate the efficiency of our NR methods.

☆ The work of the first and second authors is partially supported by National Key R&D Program of China (Grant No. 2022YFA1004500). The work of the third author is partially supported by Shenzhen Science and Technology Program (Grant No. RCJC20221008092757098) and National Natural Science Foundation of China (Grant No. 12171227).

* Corresponding author.
E-mail addresses: caichaoyi@stu.xmu.edu.cn (C. Cai), jxqiu@xmu.edu.cn (J. Qiu), wukl@sustech.edu.cn (K. Wu).

https://doi.org/10.1016/j.jcp.2023.112669
Received 23 May 2023; Received in revised form 24 October 2023; Accepted 22 November 2023

Available online 28 November 2023
0021-9991/

1. Introduction

Relativistic fluid flows widely appear in many astrophysical phenomena, such as gamma-ray bursts, supernova explosions, extragalactic jets, and accretion onto black holes. When the velocity of fluid is close to the speed of light, the classic compressible Euler equations are no longer valid due to the relativistic effect. The governing equations of the d-dimensional special relativistic hydrodynamics (RHD) can be written into a system of conservation laws as follows

$$\frac{\partial \boldsymbol{U}}{\partial t} + \sum_{i=1}^{d} \frac{\partial \boldsymbol{F}_i(\boldsymbol{U})}{\partial x_i} = \boldsymbol{0}, \tag{1.1}$$

where the conservative vector and flux vectors are given by

$$\boldsymbol{U} = \left(D, m_1, \dots, m_d, E\right)^{\top}, \tag{1.2}$$

$$\boldsymbol{F}_i = \left(Dv_i, m_1 v_i + p\delta_{1,i}, \dots, m_d v_i + p\delta_{d,i}, m_i\right)^{\top}. \tag{1.3}$$

The conservative variables $D, \boldsymbol{m} = (m_1, \dots, m_d)$, and E represent the mass density, the momentum vector, and the total energy, respectively. Let $\boldsymbol{Q} = (\rho, \boldsymbol{v}, p)^{\top}$ denote the primitive variable vector, where $\boldsymbol{v} = (v_1, \dots, v_d)$ is the fluid velocity vector, and ρ, p represent the rest-mass density and the kinetic pressure, respectively. Then $\boldsymbol{U}$ can be calculated from $\boldsymbol{Q}$ by

$$\begin{cases} D = \rho W, \\ \boldsymbol{m} = DhW\boldsymbol{v}, \\ E = DhW - p, \\ h = 1 + e + \dfrac{p}{\rho}, \\ W = 1/\sqrt{1 - |\boldsymbol{v}|^2}, \end{cases} \tag{1.4}$$

along with the ideal equation of state (EOS) considered in this paper:

$$e = \frac{p}{(\gamma - 1)\rho},$$

where $\gamma \in (1, 2]$ is the adiabatic index. In (1.4), the velocity is normalized such that the speed of light is 1, W denotes the Lorentz factor, h is the specific enthalpy, and e represents the specific internal energy. As seen from (1.4) and (1.3), both the conservative vector $\boldsymbol{U}$ and the flux $\boldsymbol{F}_i$ are explicit functions of $\boldsymbol{Q}$. However, neither $\boldsymbol{F}_i$ nor $\boldsymbol{Q}$ can be explicitly expressed by $\boldsymbol{U}$. In the computations, in order to evaluate $\boldsymbol{F}_i(\mathbf{U})$ and the eigenvalues/eigenvectors of its Jacobian matrix $\frac{\partial \boldsymbol{F}_i(\mathbf{U})}{\partial \boldsymbol{U}}$, we have to first recover the corresponding primitive variables $\boldsymbol{Q}$ from $\mathbf{U}$. This recovery procedure is complicated and typically requires one to numerically solve nonlinear algebraic equations. Several recovery algorithms were proposed in the past decades. Most algorithms calculate one intermediate variable first and then compute other primitive variables using this intermediate variable. An algorithm based on the baryon number density as the intermediate variable was constructed in [6]. Several Newton–Raphson (NR) and analytical algorithms were proposed in [37] with a variety of intermediate variables such as the pressure, the velocity, and the Lorentz factor. In [38], three analytical algorithms were designed for recovering the primitive variables for three different EOS. However, the convergence of existing NR algorithms is not guaranteed in theory, while the analytical algorithms often suffer from low accuracy and high computational cost. Recently, three robust linearly convergent iterative recovery algorithms were studied in [4].

It is often extremely difficult to obtain the analytical solutions of the RHD system (1.1) due to the high nonlinearity. As a result, numerical simulation has become an effective and practical approach to study RHD. Various numerical methods have been developed for solving the RHD equations over the past few decades, including but not limited to finite difference methods [44,6,57,35,15,50], finite volume methods [28,42,1], discontinuous Galerkin (DG) methods [34,60,43,19], and so on. The interested readers are also referred to the review papers [11,25,26] for more related developments in this direction.

The RHD equations (1.1) are a nonlinear hyperbolic system of conservation laws, which can result in discontinuities in the entropy solution even with smooth initial conditions. As well-known, this type of equations are difficult to solve due to the possibility of generating numerical oscillations near the discontinuities. Series of high-order numerical schemes based on essentially non-oscillatory (ENO) and weighted ENO (WENO) methods have been developed for solving such hyperbolic conservation laws. The history of ENO and WENO schemes can be traced back to 1985, when Harten introduced the total variation diminishing (TVD) concept [12], which formed the basis of the ENO schemes [13,14]. In 1994, Liu, Osher, and Chan proposed the first WENO scheme [24]. Jiang and Shu then improved upon it by giving the framework for the design of the smooth indicators and nonlinear weights in 1996 [18]. This kind of nonlinear weights enable the attainment of uniformly higher-order accuracy in smooth solutions, while simultaneously avoiding the emergence of numerical oscillations in discontinuous solutions. Since then, a lot of researches have sprung up on WENO schemes, including but not limited to [16,21,39,3]. Recently, Zhu and Qiu [64] proposed a simpler WENO construction, which is a convex combination of a fourth-degree polynomial and two linear polynomials with any three positive linear weights that sum to one. To address the issue of wide stencils in WENO schemes, Qiu and Shu developed Hermite WENO (HWENO) schemes [32,33], which can achieve higher-order accuracy with the same reconstruction stencils as WENO schemes. To reduce computational cost in WENO reconstruction, several hybrid WENO schemes [5,22,63] have been proposed. These schemes use linear schemes directly in smooth regions while still utilizing WENO schemes in the discontinuous regions. Recently, Zhao, Chen, and Qiu proposed several new finite

volume HWENO schemes [61,62]. Among them, the hybrid HWENO schemes [62] incorporate the thoughts behind hybrid schemes and the limiters in DG methods, making this kind of schemes more efficient and easier to implement. Compared to traditional WENO schemes, the HWENO schemes possess several advantages, such as more compact stencils in reconstructions, smaller numerical errors in smooth regions, and occasionally, higher resolution for non-smooth solutions.

Although the WENO and HWENO schemes are stable in many numerical simulations, they are generally not physical-constraint-preserving (PCP), namely, they do not always preserve the intrinsic physical constraints: for the RHD equations (1.1), such constraints include the positivity of pressure and rest-mass density, as well as the subluminal constraint on the fluid velocity. For the RHD equations (1.1), all the admissible states $\boldsymbol{U}$ satisfying these constraints form the following set

$$\mathcal{G}_0 = \left\{ \boldsymbol{U} = (D, \boldsymbol{m}, E)^\top : \ \rho(\boldsymbol{U}) > 0, \ p(\boldsymbol{U}) > 0, \ |\boldsymbol{v}(\boldsymbol{U})| < 1 \right\}. \tag{1.5}$$

In fact, preserving the numerical solutions in this set $\mathcal{G}_0$ is essential, because if any of these constraints are violated in the numerical computations, the corresponding discrete equations would become ill-posed and the simulation would break down. As we mentioned, in the RHD case, the primitive variables cannot be explicitly expressed by $\boldsymbol{U}$, so that the three functions $\rho(\boldsymbol{U})$, $p(\boldsymbol{U})$, and $\boldsymbol{v}(\boldsymbol{U})$ in (1.5) are implicit. This makes the study of PCP schemes for RHD nontrivial and more difficult than the non-relativistic hydrodynamics. In recent years, there are lots of efforts on developing high-order PCP or bound-preserving schemes for hyperbolic conservation laws via two types of limiters. The first type of limiter can be used in the finite volume and DG frameworks and was first proposed by Zhang and Shu to keep the maximum-principle-preserving property for scalar conservation laws [58] and the positivity-preserving property for the non-relativistic Euler equations [59]. The second type of limiter is a flux-correction limiter that modifies the high-order flux with a first-order PCP flux to obtain a new flux with high accuracy and PCP property; cf. [54,53,17]. The interested readers are referred to the reviews [55,40] for more related works.

The first PCP work on RHD was made in [50], which provided a rigorous proof for the PCP property of the local Lax–Friedrichs flux and proposed the PCP finite difference WENO schemes for RHD. The following explicit equivalent form of the admissible state set (1.5) was also proved in [50]

$$\mathcal{G} = \left\{ \boldsymbol{U} = (D, \boldsymbol{m}, E)^\top : \ D > 0, \ g(\boldsymbol{U}) := E - \sqrt{D^2 + |\boldsymbol{m}|^2} > 0 \right\}, \tag{1.6}$$

where $g(\boldsymbol{U})$ is a concave function, and moreover, $\mathcal{G} = \mathcal{G}_0$ is a convex set. Qin, Shu, and Yang developed a bound-preserving DG method for RHD in [31]. The PCP Lagrangian finite volume schemes were designed in [23]. A PCP central DG method was proposed for RHD with a general EOS in [51]. The framework for designing provably high-order PCP methods for general RHD was established in [45]. More recently, a minimum principle on specific entropy and high-order accurate invariant region preserving numerical methods were studied for RHD in [47]. A PCP finite volume WENO method was developed on unstructured meshes in [4], where three robust algorithms were also introduced for recovering the primitive variables. Besides, the PCP schemes were also studied for the relativistic magnetohydrodynamics (MHD) in [52,48]. Most notably, the theoretical analyses in [52,46] based on the geometric quasilinearization technique [49] revealed that the PCP property of MHD schemes is strongly connected with a discrete divergence-free condition on the magnetic field. In addition, a flux limiter was proposed in [36] to preserve the positivity of rest-mass density, and a subluminal reconstruction technique was designed for relativistic MHD in [1]. As we have mentioned, neither the flux $\boldsymbol{F}_i$ nor $\boldsymbol{Q}$ can be explicitly expressed by $\boldsymbol{U}$ in the RHD case. In order to evaluate the flux and the eigenvalues/eigenvectors of its Jacobian matrix, we have to recover the primitive variables $\boldsymbol{Q}$ from $\mathbf{U}$. This recovery procedure requires solving a nonlinear algebraic equation by some root-finding algorithms. Although the PCP property $\boldsymbol{U} \in \mathcal{G}$ guarantees the existence and uniqueness of the corresponding physical primitive variables in theory [50], it however does not ensure the convergence of the root-finding algorithms, nor the physical constraints of the computed primitive variables obtained by the root-finding algorithms.

This paper aims to design genuinely PCP schemes. We first propose three efficient PCP NR methods for robustly recovering primitive variables from conservative variables. Importantly, we rigorously prove that these NR methods are always convergent and PCP, meaning they preserve the physical constraints throughout the NR iterations. The discovery of these robust NR methods and their PCP convergence analyses are highly nontrivial and become the most significant contribution of this work. In particular, our analyses involve careful and detailed investigations of the convexity/concavity structures of the iterative functions. The proposed NR methods are versatile and can be integrated with any RHD schemes requiring the recovery of primitive variables, including but not limited to TVD, WENO, HWENO, and DG schemes for RHD. As an application, we apply our NR methods to develop robust and efficient high-order PCP finite volume HWENO schemes for the RHD equations (1.1). Our approach builds upon the NR methods, the hybrid high-order HWENO reconstruction proposed in [62], a PCP limiter, and strong-stability-preserving Runge–Kutta method for time discretization. We rigorously prove the PCP property of our HWENO schemes under a CFL condition, by using the Lax–Friedrichs splitting property and convex decomposition techniques. Moreover, we suggest the rescaled eigenvectors for characteristic decomposition and the scale-invariant nonlinear weights to address the issue of numerical oscillations arising from the wide range of variable spans in the RHD equations and enhance the performance of the HWENO schemes in simulating large-scale RHD problems. We implement the proposed one-dimensional (1D) and two-dimensional (2D) PCP HWENO schemes, and provide extensive challenging numerical tests to demonstrate the robustness, accuracy, and high resolution of our PCP HWENO schemes and to validate the efficiency of our NR methods.

This paper is organized as follows. Section 2 proposes three efficient PCP convergent NR methods for recovering primitive variables and provides the theoretical analysis on their convergence and PCP property. Section 3 introduces the 1D PCP finite volume HWENO method for RHD and provides the theoretical analysis of the PCP property. Section 4 extends the method and analysis to the RHD systems in two dimensions and the cylindrical coordinates. Section 5 conducts several numerical experiments

to demonstrate the PCP property, accuracy, and effectiveness of the PCP HWENO schemes and NR methods. Section 6 gives the conclusion of this paper.

2. Efficient PCP convergent Newton–Raphson methods for recovering primitive variables

In sections 3 and 4, we will develop a PCP HWENO method that ensures the numerical solutions of the conservative variables $\mathbf{U}$ in the admissible state set $\mathcal{G}$. According to the analysis in [50], when $\boldsymbol{U} \in \mathcal{G}$, the corresponding primitive variables $\boldsymbol{Q} = (\rho, \boldsymbol{v}, p)^{\top}$ are uniquely determined and satisfy the physical constraints

$$\rho > 0, \quad p > 0, \quad |\boldsymbol{v}| < 1. \tag{2.1}$$

When computing the fluxes $\boldsymbol{F}_i(\boldsymbol{U})$ with the conservative vectors $\boldsymbol{U} \in \mathcal{G}$, it is necessary to recover the primitive variables $\boldsymbol{Q} = (\rho, \boldsymbol{v}, p)^{\top}$ from $\boldsymbol{U}$. This recovery procedure requires solving a nonlinear algebraic equation by some root-finding algorithms, because there is no explicit expressions for $\boldsymbol{Q}$ in terms of $\boldsymbol{U}$. In the past decades, many algorithms recovering the primitive variables were proposed; see [37] for a review.

The PCP property $\boldsymbol{U} \in \mathcal{G}$ guarantees the uniqueness of the corresponding physical primitive variables in theory. However, it does not ensure the convergence of the root-finding algorithms, nor the physical constraints (2.1) for the primitive variables computed by the root-finding algorithms. We would like to seek iterative root-finding algorithms that are convergent and PCP, namely, preserve the physical constraints (2.1) during the iteration process. In particular, we are interested in recovering the pressure $p(\boldsymbol{U})$ first; once $p(\boldsymbol{U})$ is recovered, the velocity vector and the density can be calculated sequentially as follows:

$$\boldsymbol{v}(\boldsymbol{U}) = \boldsymbol{m}/(E + p(\boldsymbol{U})), \qquad \rho(\boldsymbol{U}) = D\sqrt{1 - |\boldsymbol{v}(\boldsymbol{U})|^2}. \tag{2.2}$$

We observe that, for a given $\boldsymbol{U} \in \mathcal{G}$ satisfying $D > 0$ and $E > \sqrt{D^2 + |\boldsymbol{m}|^2}$, if the recovered $p(\boldsymbol{U}) > 0$, then the calculation (2.2) leads to $|\boldsymbol{v}(\boldsymbol{U})| < 1$ and $\rho(\boldsymbol{U}) > 0$. Hence we propose the following definitions.

Definition 2.1. Given $\boldsymbol{U} \in \mathcal{G}$, a pressure-recovering algorithm is called PCP, if all the approximate pressures in the iterative sequence $\{p_n\}_{n\geq 1}$ are always positive.

Definition 2.2. Given $\boldsymbol{U} \in \mathcal{G}$, a pressure-recovering algorithm is called convergent, if the iterative sequence $\{p_n\}_{n\geq 1}$ converges to the physical pressure $p(\boldsymbol{U})$, namely, $\lim_{n\to+\infty} p_n = p(\boldsymbol{U})$.

In [4], three PCP convergent pressure-recovering algorithms were developed. Those algorithms are based on bisection, fixed-point iteration, and a hybrid iteration combining them, respectively. Consequently, those algorithms in [4] have only a first order of linear convergence. In this section, we develop three faster pressure-recovering algorithms, which are Newton–Raphson (NR) method and have a convergence order of 2. Furthermore, we will rigorously prove that the three proposed NR methods are both PCP and convergent.

2.1. NR-I method: monotonically convergent PCP NR iteration

As shown in [50], the true physical pressure $p(\boldsymbol{U})$ corresponding to a conservative vector $\boldsymbol{U} = (D, \boldsymbol{m}, E)^{\top}$ satisfies the following nonlinear algebraic equation:

$$\Phi(p) := \frac{p}{\gamma - 1} - E + \frac{|\boldsymbol{m}|^2}{E + p} + D\sqrt{1 - \frac{|\boldsymbol{m}|^2}{(E + p)^2}} = 0. \tag{2.3}$$

When $\boldsymbol{U} \in \mathcal{G}$, we can show that the function $\Phi(p)$ is strictly monotonically increasing with respect to $p \in [0, +\infty)$. Moreover, $\Phi(0) < 0$ and $\lim_{p\to+\infty} \Phi(p) = +\infty$. This yields the equation (2.3) admits a unique positive solution, which is the true physical pressure $p(\boldsymbol{U})$. In addition, the equation (2.3) implies that the true physical pressure $p(\boldsymbol{U})$ satisfies

$$\frac{p}{\gamma - 1} - E + \frac{|\boldsymbol{m}|^2}{E + p} < 0. \tag{2.4}$$

A natural idea is to directly solve equation (2.3) by using the NR method. Unfortunately, the numerical experiments in [4] indicate that the convergence of such a NR method requires a good initial guess, and the approximate pressure may become negative during the NR iterations.

In order to introduce our monotonically convergent NR method, we define

$$h_1(p) := \left(|\boldsymbol{m}|^2 + (E + p)\left(\frac{p}{\gamma - 1} - E\right)\right)^2, \tag{2.5}$$

$$h_2(p) := D^2\left((E + p)^2 - |\boldsymbol{m}|^2\right). \tag{2.6}$$

After a simple transformation of (2.3), we obtain a quartic equation of p:

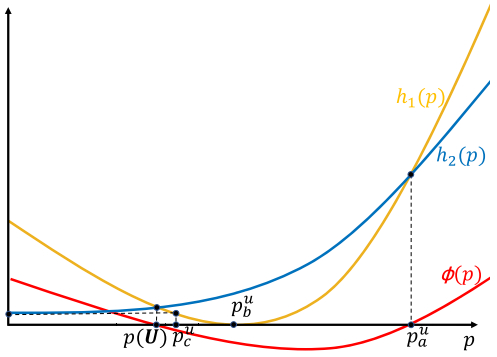


Fig. 1. The graphs of $h_1(p)$, $h_2(p)$, and $\phi(p)$.

$$\phi(p) := (\gamma-1)^2\left[h_1(p)-h_2(p)\right] = c_0 + c_1 p + c_2 p^2 + c_3 p^3 + p^4 = 0 \tag{2.7}$$

with

$$\begin{cases} c_0 = (|\boldsymbol{m}|^2 - E^2)(|\boldsymbol{m}|^2 - E^2 + D^2)(\gamma-1)^2, \\ c_1 = 2E(2-\gamma)(|\boldsymbol{m}|^2 - E^2)(\gamma-1) - 2ED^2(\gamma-1)^2, \\ c_2 = E^2(\gamma^2 - 6\gamma + 6) + 2|\boldsymbol{m}|^2(\gamma-1) - D^2(\gamma-1)^2, \\ c_3 = 2E(2-\gamma). \end{cases} \tag{2.8}$$

The graphs of $h_1(p)$, $h_2(p)$, and $\phi(p)$ are shown in Fig. 1.

The transformation from (2.3) to (2.7) produces some additional (nonphysical) roots, which fail to meet the constraints $p>0$ and (2.4). In the following, we will prove that the minimum positive root of equation (2.7) corresponds to the physical pressure $p(\boldsymbol{U})$. Furthermore, we will propose a practical NR method that are highly efficient and can be proven to monotonically converge to the physical pressure $p(\boldsymbol{U})$.

Lemma 2.1. *Given $\boldsymbol{U} \in \mathcal{G}$, we have $c_0 > 0$, $c_1 < 0$, and $c_3 \geq 0$. Furthermore, $c_3 = 0$ if and only if $\gamma = 2$.*

Proof. Since $D>0$, $E > \sqrt{D^2 + |\boldsymbol{m}|^2}$, and $\gamma \in (1,2]$, we have

(i) $c_0 = (E^2 - |\boldsymbol{m}|^2)(E^2 - D^2 - |\boldsymbol{m}|^2)(\gamma-1)^2 > 0$.
(ii) $c_1 = 2E(2-\gamma)(|\boldsymbol{m}|^2 - E^2)(\gamma-1) - 2ED^2(\gamma-1)^2 < -2ED^2(\gamma-1)^2 < 0$.
(iii) $c_3 = 2E(2-\gamma) \geq 0$, and the equality holds if and only if $\gamma = 2$. □

Lemma 2.2. *Given $\boldsymbol{U} \in \mathcal{G}$, $\phi(p)$ has at least two different positive roots, which are located in the intervals $(0, p_b^u)$ and $(p_b^u, +\infty)$, respectively, where*

$$p_b^u := \frac{1}{2}\left(E(\gamma-2) + \sqrt{E^2(2-\gamma)^2 - 4(\gamma-1)(|\boldsymbol{m}|^2 - E^2)}\right). \tag{2.9}$$

Proof. Consider the quadratic function $h_3(p) = |\boldsymbol{m}|^2 + (E+p)\left(\frac{p}{\gamma-1} - E\right)$. Note that $h_3(0) = |\boldsymbol{m}|^2 - E^2 < 0$, and $h_3'(p) = \frac{2p}{\gamma-1} + \frac{2-\gamma}{\gamma-1}E > 0$ when $p>0$. Thus $h_3(p)$ is strictly increasing on $[0,+\infty)$ and has only one positive root, which is exactly p_b^u defined in (2.9). Note that $h_2(p)$ is monotonically increasing when $p \geq 0$, which implies

$$h_2(p) \geq h_2(0) = D^2(E^2 - |\boldsymbol{m}|^2) > D^4 > 0, \quad \forall p \geq 0.$$

Hence we have

$$\phi(p_b^u) = (\gamma-1)^2(h_1(p_b^u) - h_2(p_b^u)) = (\gamma-1)^2(h_3^2(p_b^u) - h_2(p_b^u)) = -(\gamma-1)^2 h_2(p_b^u) < 0.$$

Since $\phi(0) = c_0 > 0$ and $\lim_{p\to+\infty}\phi(p) = +\infty$, according to the zero point theorem, we know that $\phi(p)$ has at least two different positive roots, which are located in the intervals $(0, p_b^u)$ and $(p_b^u, +\infty)$, respectively. The proof is completed. □

For convenience, we will count the number of roots by including the multiplicity of a repeated root, unless otherwise specified.

Lemma 2.3. *If $\phi(p)$ has 4 real roots, then it has 2 negative roots.*

Proof. Assume the 4 real roots of $\phi(p)$ are $\hat{p}_1 \le \hat{p}_2 \le \hat{p}_3 \le \hat{p}_4$. According to Lemma 2.2, we have $\hat{p}_4 > \hat{p}_3 > 0$. Then it suffices to prove $\hat{p}_1 < \hat{p}_2 < 0$. According to Vieta's formulas, $\hat{p}_1 + \hat{p}_2 + \hat{p}_3 + \hat{p}_4 = -c_3 \le 0$ and $\hat{p}_1\hat{p}_2\hat{p}_3\hat{p}_4 = c_0 > 0$, we can conclude that $\hat{p}_1 \le \hat{p}_2 < 0$, which finishes the proof. □

With Lemmas 2.2 and 2.3, we immediately obtain the following theorem.

Theorem 2.1. *The quartic polynomial $\phi(p)$ has either 2 different positive roots and 2 negative roots, or 2 different positive roots and 2 complex roots.*

Theorem 2.2. *The smallest positive root of $\phi(p)$ is the unique positive root of equation (2.3), which is the physical pressure $p(\boldsymbol{U})$.*

Proof. Denote the larger positive root of $\phi(p)$ as p_a^μ. The proof of Lemma 2.2 implies that $p_a^\mu > p_b^\mu > 0$. It suffices to prove that $\Phi(p_a^\mu) \ne 0$. Recall that $h_3(p)$ is strictly increasing in the interval $[0, +\infty)$. Therefore,

$$\begin{aligned}\Phi(p_a^\mu) &= \frac{p_a^\mu}{\gamma - 1} - E + \frac{|\boldsymbol{m}|^2}{E + p_a^\mu} + D\sqrt{1 - \frac{|\boldsymbol{m}|^2}{(E + p_a^\mu)^2}} \\ &= \frac{h_3(p_a^\mu)}{E + p_a^\mu} + D\sqrt{1 - \frac{|\boldsymbol{m}|^2}{(E + p_a^\mu)^2}} \\ &> \frac{h_3(p_b^\mu)}{E + p_a^\mu} + D\sqrt{1 - \frac{|\boldsymbol{m}|^2}{(E + p_a^\mu)^2}} = D\sqrt{1 - \frac{|\boldsymbol{m}|^2}{(E + p_a^\mu)^2}} > 0,\end{aligned}$$

which finishes the proof. □

The relative positions of $p(\boldsymbol{U})$, p_a^μ, and p_b^μ are illustrated in Fig. 1.

Before giving our NR-I method, we present several lemmas, which are useful for establishing the convergence and PCP property of the NR-I method.

Lemma 2.4. *Let $\{p_n\}_{n\ge 0}$ denote the iteration sequence obtained using the NR method to solve an equation $f(p) = 0$. We assume that p_* is a root of $f(p) = 0$. If $p_0 < p_*$, $f \in C^2[p_0, p_*)$, and one of the following two conditions holds for all $p \in [p_0, p_*)$:*

(i) $f'(p) < 0, f''(p) \ge 0$,
(ii) $f'(p) > 0, f''(p) \le 0$,

then the NR iteration sequence $\{p_n\}_{n\ge 0}$ is monotonically increasing and converges to p_*.

Proof. We only show the proof under the condition (i), while the proof under condition (ii) is similar and thus omitted.

Firstly, we prove that $\{p_n\}_{n\ge 0}$ is monotonically increasing and that p_* is an upper bound of $\{p_n\}_{n\ge 0}$. It suffices to prove that $p_* - p_{n+1} = p_* - (p_n - \frac{f(p_n)}{f'(p_n)}) \ge 0$ and $p_{n+1} - p_n = -\frac{f(p_n)}{f'(p_n)} > 0$ if $p_0 \le p_n < p_*$. Under the condition (i), we have $f(p_n) > f(p_*) = 0$ and $f'(p_n) < 0$, so that $p_{n+1} - p_n = -\frac{f(p_n)}{f'(p_n)} > 0$. Define $f_1(p) := p_* - (p - \frac{f(p)}{f'(p)})$, then

$$f_1'(p) = -1 + \frac{f'(p)^2 - f(p)f''(p)}{f'(p)^2} = -\frac{f(p)f''(p)}{f'(p)^2} \le 0, \qquad \forall p \in [p_0, p_*).$$

Since $f_1(p_*) = 0$ and $p_n < p_*$, we have

$$0 = f_1(p_*) \le f_1(p_n) = p_* - \left(p_n - \frac{f(p_n)}{f'(p_n)}\right) = p_* - p_{n+1}.$$

Secondly, we prove the convergence. Since $\{p_n\}_{n\ge 0}$ is monotonically increasing and has an upper bound p_*, by the monotone convergence theorem, we know that the sequence $\{p_n\}_{n\ge 0}$ has a limit $p_{**} \le p_*$. Therefore,

$$0 = \lim_{n\to+\infty}(p_{n+1} - p_n) = -\lim_{n\to+\infty}\frac{f(p_n)}{f'(p_n)} = -\frac{f(p_{**})}{f'(p_{**})},$$

which implies $f(p_{**}) = 0$. Since $p_0 \le p_{**} \le p_*$ and $f(p) > 0$ when $p \in [p_0, p_*)$, we obtain $p_{**} = p_*$. Hence $\lim_{n\to+\infty} p_n = p_{**} = p_*$. The proof is completed. □

Similarly, we have the following lemma.

Lemma 2.5. *Let $\{p_n\}_{n\geq 0}$ denote the iteration sequence obtained using the NR method to solve an equation $f(p)=0$. We assume that p_* is a root of $f(p)=0$. If $p_0 > p_*$, $f \in C^2(p_*, p_0]$, and one of the following two conditions holds for all $p \in (p_*, p_0]$:*

(i) $f'(p) < 0, f''(p) \leq 0$,
(ii) $f'(p) > 0, f''(p) \geq 0$,

the NR iteration sequence $\{p_n\}_{n\geq 0}$ is monotonically decreasing and converges to p_*.

Theorem 2.3. *If $\phi''(0) = 2c_2 > 0$, then $\phi''(p) > 0$ and $\phi'(p) < 0$ for all $p \in [0, p(\boldsymbol{U}))$.*

Proof. Note that $\phi''(p) = 12p^2 + 6c_3 p + 2c_2$ and $\phi'''(p) = 24p + 6c_3$. Because $c_3 \geq 0$, we have $\phi'''(p) > 0$ when $p > 0$, which yields $\phi''(p)$ is strictly increasing in the interval $[0, +\infty)$. Since $\phi''(0) = 2c_2 > 0$, we have $\phi''(p) \geq \phi''(0) > 0$ for all $p \in [0, p(\boldsymbol{U})) \subset [0, +\infty)$.

We then show $\phi'(p) < 0$ using proof by contradiction. Suppose that there exists $p_a \in [0, p(\boldsymbol{U}))$ such that $\phi'(p_a) \geq 0$. Because $\phi'(0) = c_1 < 0$, there must exist $p_b \in (0, p_a]$ such that $\phi'(p_b) = 0$ by the intermediate value theorem. Since $p_a^u > p(\boldsymbol{U})$ and $\phi(p(\boldsymbol{U})) = \phi(p_a^u) = 0$, by Rolle's theorem, there exists $p_c \in [p(\boldsymbol{U}), p_a^u]$, such that $\phi'(p_c) = 0$. Therefore, $p_b \leq p_a < p(\boldsymbol{U}) \leq p_c$. Since $\phi'(p_b) = \phi'(p_c) = 0$, there exists $p_d \in [p_b, p_c] \subset [0, +\infty)$, such that $\phi''(p_d) = 0$, which is contradictory to $\phi''(p) > 0$ for all $p \in [0, +\infty)$. Thus, the assumption is incorrect, and we have $\phi'(p) < 0$ for all $p \in [0, p(\boldsymbol{U}))$. The proof is completed. □

Theorem 2.4. *If $\phi''(0) = 2c_2 \leq 0$, then the largest root of the quadratic polynomial $\phi''(p)$, denoted by p_e, satisfies $p_e = \frac{-3c_3 + \sqrt{9c_3^2 - 24c_2}}{12} \geq 0$. Furthermore, we have*

(i) *If $0 \leq p_e < p(\boldsymbol{U})$, then $\phi''(p) \geq 0$ and $\phi'(p) < 0$ for all $p \in [p_e, p(\boldsymbol{U}))$.*
(ii) *If $p_e > p(\boldsymbol{U})$, then $\phi''(p) \leq 0$ and $\phi'(p) < 0$ for all $p \in (p(\boldsymbol{U}), p_e]$.*

Proof. Recall that $\phi''(p) = 12p^2 + 6c_3 p + 2c_2$, and $\phi'''(p) = 24p + 6c_3 > 0$ when $p > 0$, which yields $\phi''(p)$ is strictly increasing in $[0, +\infty)$. Note that $\lim_{p \to +\infty} \phi''(p) = +\infty$. If $\phi''(0) = 2c_2 \leq 0$, then by the intermediate value theorem, we know $p_e = \frac{-3c_3 + \sqrt{9c_3^2 - 24c_2}}{12} \geq 0$.

We then prove the conclusions (i) and (ii) separately.

(i) It is obvious that $\phi''(p) \geq 0$ when $p \geq p_e$. Suppose there exists $p_a \in [p_e, p(\boldsymbol{U}))$ such that $\phi'(p_a) \geq 0$, then $\phi'(p) \geq 0$ for all $p \in [p_a, p(\boldsymbol{U}))$ because $\phi''(p) \geq 0$ for all $p \in [p_e, p(\boldsymbol{U}))$. This means $\phi'(p)$ is monotonically increasing in $[p_a, p(\boldsymbol{U}))$, implying $\phi(p_a) \leq \phi(p(\boldsymbol{U})) = 0$. Since $\phi(0) > 0$, according to the intermediate value theorem, there exists $p_b \in (0, p_a]$ such that $\phi(p_b) = 0$. This contradicts with the fact that $p(\boldsymbol{U})$ is the smallest positive root of $\phi(p)$. Thus, the assumption is incorrect, and we have $\phi'(p) < 0$ for all $p \in [p_e, p(\boldsymbol{U}))$.
(ii) Notice that $\phi''(p) \leq 0$ for all $p \in (p(\boldsymbol{U}), p_e] \subset [0, p_e]$. Since $\phi'(0) = a_1 < 0$ and $\phi''(p) \leq 0$ when $p \in [0, p_e]$, we have $\phi'(p) \leq \phi'(0) < 0$ when $p \in [0, p_e] \supset (p(\boldsymbol{U}), p_e]$.

The proof is completed. □

Inspired by Theorems 2.3 and 2.4 as well as Lemmas 2.4 and 2.5, we design the following NR method for recovering the pressure $p(\mathbf{U})$ from the conservative vector $\boldsymbol{U}$.

Algorithm 2.1 *(NR-I method).* The NR iteration reads

$$p_{n+1} = p_n - \frac{\phi(p_n)}{\phi'(p_n)},$$

with p_0 given by

$$p_0 = \begin{cases} 0, & \text{if } c_2 > 0, \\ \frac{-3c_3 + \sqrt{9c_3^2 - 24c_2}}{12}, & \text{otherwise.} \end{cases} \tag{2.10}$$

The specific expressions for $\phi(p)$ and $c_i, i = 2, 3$, are given in (2.7) and (2.8).

Note that $p_0 \geq 0$ and $p(\boldsymbol{U}) > 0$. We immediately obtain the following conclusions from Theorems 2.3 and 2.4 as well as Lemmas 2.4 and 2.5.

Theorem 2.5. *The iteration sequence $\{p_n\}_{n\geq 0}$ generated by Algorithm 2.1 converges monotonically to $p(\boldsymbol{U})$. Furthermore, Algorithm 2.1 is PCP.*

Theorem 2.6. *Algorithm 2.1 has a quadratic convergence.*

Proof. Thanks to Theorems 2.1 and 2.2, we know that the physical pressure $p(\mathbf{U})$, as a root of $\phi(p)$, is not a repeated root, namely, its multiplicity is one. Therefore, as a NR iteration, Algorithm 2.1 has quadratic convergence. □

Remark 2.1. In practical computations, the Horner's rule can be applied to efficiently evaluate the polynomials $\phi(p)$ and $\phi'(p)$.

Remark 2.2. When using the NR method to solve $\phi(p) = 0$, oscillations may occur when the value of $\phi(p_n)$ is very close to zero; see [10]. Detecting the oscillations caused by round-off errors and then stopping the iteration are important to avoid unnecessary computational costs. Since Algorithm 2.1 has monotonic convergence (Theorem 2.5), it is reasonable to expect that oscillations occur when the theoretical monotonicity of the iteration sequence is lost due to round-off errors. Therefore, in addition to the standard stopping criterion $|\phi(p_n)| < \epsilon_{target}$ (ϵ_{target} denotes the target accuracy), we also include such oscillations as a termination criterion in our computations.

2.2. Analytical expression of $p(\boldsymbol{U})$

With Lemma 2.1 and Theorems 2.1–2.2, we can obtain the analytical expression of pressure $p(\boldsymbol{U})$ by using the Ferrari method.

Algorithm 2.2 *(Analytical).* $p(\boldsymbol{U}) = \frac{M_5 - c_3 - M_6}{4}$ with

$$\begin{cases} M_1 = \dfrac{c_2^2 + 12c_0 - 3c_3c_1}{9}, \\ M_2 = \dfrac{27c_1^2 + 2c_2^3 + 27c_3^2c_0 - 72c_2c_0 - 9c_3c_2c_1}{54}, \\ M_3 = \left(M_2 + (M_2^2 - M_1^3)^{\frac{1}{2}} \right)^{\frac{1}{3}}, \\ M_4 = M_3 + \dfrac{M_1}{M_3} + \dfrac{c_2}{3}, \\ M_5 = (c_3^2 + 4(M_4 - c_2))^{\frac{1}{2}}, \\ M_6 = \left((c_3 - M_5)^2 - 8\left(M_4 - \dfrac{c_3 M_4 - 2c_1}{M_5} \right) \right)^{\frac{1}{2}}, \end{cases}$$

where the specific expressions of c_k $(k = 0, 1, 2, 3)$ are given in (2.8). The function $(\cdot)^{\frac{1}{n}}$ $(n = 2, 3)$ is a single-valued complex function defined as follows

$$(A_1 + A_2 i)^{\frac{1}{n}} = \left(A^{\frac{1}{n}} \cos\frac{\theta}{n} \right) + \left(A^{\frac{1}{n}} \sin\frac{\theta}{n} \right) i, \tag{2.11}$$

where $i = \sqrt{-1}$ is the imaginary unit, $A_1, A_2 \in \mathbb{R}$ and

$$A = \sqrt{A_1^2 + A_2^2}, \tag{2.12}$$

$$\theta = \arctan\frac{A_2}{A_1}. \tag{2.13}$$

It can be seen that $p(\boldsymbol{U})$ is essentially expressed explicitly in terms of $\boldsymbol{U}$. However, because the $(\cdot)^{\frac{1}{n}}$ operation $(n = 2, 3)$ inherently includes trigonometric, inverse trigonometric, and cube root or square root operations, using Algorithm 2.2 to calculate $p(\boldsymbol{U})$ is very costly and sometimes of low accuracy. We will observe this fact and compare Algorithm 2.2 with our NR methods in the numerical experiments in Section 5.

2.3. NR-II method: robust PCP convergent NR iteration

To facilitate the algorithm design, we define

$$\psi(p) := (E + p)\Phi(p) = |\boldsymbol{m}|^2 + (E + p)\left(\frac{p}{\gamma - 1} - E \right) + D\sqrt{(E + p)^2 - |\boldsymbol{m}|^2} \tag{2.14}$$

and transform the equation (2.3) into

$$\psi(p) = 0. \tag{2.15}$$

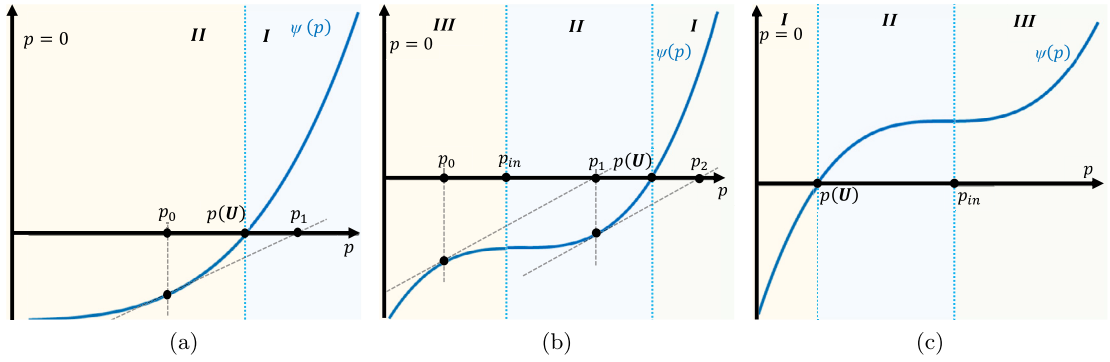


Fig. 2. Three possible cases of the concavity/convexity structure of $\psi(p)$.

In this subsection, we aim to study the NR iteration method solving the equation (2.15). In particular, we would like to find an appropriate initial value p_0, such that the resulting NR method is PCP and provably convergent. Moreover, we hope that p_0 is sufficiently close to $p(\boldsymbol{U})$ in order to reduce the number of iterations.

Define

$$p_c^u := \frac{1}{2}\left((\gamma-2)E + \sqrt{(2-\gamma)^2E^2 - 4(\gamma-1)\left[(|\boldsymbol{m}|^2 - E^2) + D\sqrt{E^2-|\boldsymbol{m}|^2}\right]}\right), \tag{2.16}$$

which is the root of the following equation

$$h_1(p_c^u) = h_2(0). \tag{2.17}$$

We observe that p_c^u is closer to $p(\boldsymbol{U})$ than p_b^u, as shown in the Fig. 1 and proven in Lemma 2.6.

Lemma 2.6. $p(\boldsymbol{U}) < p_c^u < p_b^u < p_a^u$.

Proof. According Lemma 2.2 and Theorem 2.2, we obtain $0 < p(\boldsymbol{U}) < p_b^u < p_a^u$. Recalling that $h_1(p)$ is strictly decreasing in the interval $(0, p_b^u)$ and $h_2(p)$ is strictly increasing in $(0, p_b^u)$, we have

$$h_1(p_b^u) = 0 < D^2(E^2 - |\boldsymbol{m}|^2) = h_2(0) < h_2(p(\boldsymbol{U})) = h_1(p(\boldsymbol{U})) < h_1(0).$$

By the intermediate value theorem, there exists a unique point $p_c^u \in (0, p_b^u)$ such that $h_1(p_c^u) = h_2(0)$ and $p(\boldsymbol{U}) < p_c^u < p_b^u$. The expression (2.16) of p_c^u can be easily obtained by solving the equation $h_1(p_c^u) = h_2(0)$. The proof is completed. □

Although p_c^u is close to $p(\boldsymbol{U})$, the NR method for solving the equation (2.15) with $p_0 = p_c^u$ is not always convergent and PCP. One can verify that

$$\begin{aligned}
&\psi'(p) = \frac{2p + (2-\gamma)E}{\gamma - 1} + \frac{D(E+p)}{\sqrt{(E+p)^2 - |\boldsymbol{m}|^2}} > 0, \quad \forall p \geq 0,\\
&\psi''(p) = \frac{2}{\gamma-1} - \frac{D|\boldsymbol{m}|^2}{\left[(E+p)^2 - |\boldsymbol{m}|^2\right]^{\frac{3}{2}}},\\
&\psi'''(p) > 0, \quad \forall p \geq 0,
\end{aligned} \tag{2.18}$$

and $\lim_{p\to+\infty} \psi''(p) = \frac{2}{\gamma-1} > 0$. If $\psi''(0) \geq 0$, then (2.18) implies that $\psi''(p) > 0$ when $p > 0$. If $\psi''(0) < 0$, then because $\lim_{p\to+\infty} \psi''(p) > 0$, there exists a inflection point $p_{in} > 0$ satisfying $\psi''(p_{in}) = 0$. Since $\psi'''(p) > 0$ when $p > 0$, we can deduce that $\psi''(p) > 0$ on $(p_{in}, +\infty)$ and $\psi''(p) < 0$ on $(0, p_{in})$. Therefore, the concavity/convexity structure of the function $\psi(p)$ in $[0, +\infty)$ can only have three cases:

(a) $\psi''(p) > 0$ when $p > 0$.
(b) There exists a inflection point $p_{in} \in (0, p(\boldsymbol{U}))$ satisfying $\psi''(p_{in}) = 0$. $\psi''(p) > 0$ on $(p_{in}, +\infty)$, while $\psi''(p) < 0$ on $(0, p_{in})$.
(c) There exists a inflection point $p_{in} \geq p(\boldsymbol{U})$ satisfying $\psi''(p_{in}) = 0$. $\psi''(p) > 0$ on $(p_{in}, +\infty)$, while $\psi''(p) < 0$ on $(0, p_{in})$.

The graphs of $\psi(p)$ are illustrated in Fig. 2 for Cases (a), (b), and (c), respectively.

Theorem 2.7. *In Cases (a) and (b), the NR iteration for solving $\psi(p) = 0$ always converges to $p(\boldsymbol{U})$ with any initial guess $p_0 \geq 0$. In Case (c), the NR iteration for solving $\psi(p) = 0$ always converges to $p(\boldsymbol{U})$ with any initial guess $p_0 \in [0, p(\boldsymbol{U})]$, and if $p_0 > p(\mathbf{U})$, then it may fail to converge to $p(\boldsymbol{U})$. Furthermore, if the NR iteration fails to converge, then negative p_n would appear in the iteration sequence $\{p_n\}_{n\geq 0}$.*

Proof. We discuss the three cases in Fig. 2 separately.

(a) In this case, $\psi''(p) > 0$ when $p > 0$.
(I) If $p_0 \in [p(\boldsymbol{U}), +\infty)$, then the iteration sequence $\{p_n\}_{n\geq 0}$ converges monotonically to $p(\boldsymbol{U})$ according to Lemma 2.5.
(II) If $p_0 \in [0, p(\boldsymbol{U}))$, then $p_1 = p_0 - \frac{\psi(p_0)}{\psi'(p_0)}$, $\psi(p_1) > \psi(p_0) + (p_1 - p_0)\psi'(p_0) = 0$. Thus $p_1 \in [p(\boldsymbol{U}), +\infty)$. Then, following the discussion of Case (I), the iteration sequence $\{p_n\}_{n\geq 1}$ converges monotonically to $p(\boldsymbol{U})$.

(b) In this case, $p(\boldsymbol{U}) > p_{in} > 0$, where $\psi''(p_{in}) = 0$.
(I) If $p_0 \in [p(\boldsymbol{U}), +\infty)$, then the iteration sequence $\{p_n\}_{n\geq 0}$ converges monotonically to $p(\boldsymbol{U})$ according to Lemma 2.5.
(II) If $p_0 \in [p_{in}, p(\boldsymbol{U}))$, then similarly to Case (a)(II), we have $p_1 \in [p(\boldsymbol{U}), +\infty)$, and the iteration sequence $\{p_n\}_{n\geq 1}$ converges monotonically to $p(\boldsymbol{U})$.
(III) If $p_0 \in [0, p_{in})$, $p_1 = p_0 - \frac{\psi(p_0)}{\psi'(p_0)} > p_0$. If $p_1 \geq p_{in}$, then the discussion returns to Cases (I) and (II). Thus we only need to discuss the case when $p_1 \in [0, p_{in})$. By repeatedly following the aforementioned discussions, as long as $p_n \geq p_{in}$ appears in the iteration, we can return to the discussion of Case (I) or (II), and conclude that the NR method converges. It remains to discuss whether it is possible that $p_n < p_{in}$ for all $n \geq 0$. Assume that such a situation occurs, then since $p_n < p_{in}$ and $p_{n+1} > p_n$, according to the monotone bounded convergence theorem, $\{p_n\}_{n\geq 0}$ has a limit $p^* \in [0, p_{in}]$. Therefore,

$$0 = \lim_{n\to+\infty}(p_{n+1} - p_n) = -\lim_{n\to+\infty}\frac{\psi(p_n)}{\psi'(p_n)} = -\frac{\psi(p^*)}{\psi'(p^*)},$$

which yields $\psi(p^*) = 0$. This is contradictory to $p^* \in [0, p_{in}]$ and $p_{in} < p(\boldsymbol{U})$ (note that $p(\boldsymbol{U})$ is the unique positive root of $\psi(p)$). Hence, the assumption is incorrect, and there always exists a n such that $p_n \geq p_{in}$. In short, the NR method always converges.

(c) In this case, $p_{in} \geq p(\boldsymbol{U}) > 0$ where $\psi''(p_{in}) = 0$.
(I) If $p_0 \in [0, p(\boldsymbol{U})]$, then the iteration sequence $\{p_n\}_{n\geq 0}$ converges monotonically to $p(\boldsymbol{U})$ according to Lemma 2.4.
(II) If $p_0 \in (p(\boldsymbol{U}), p_{in}]$, then similar to Case (a)(II) and Case (b)(II), we have $p_1 \leq p(\boldsymbol{U})$. If $p_1 < 0$, the convergence cannot be guaranteed. If $p_1 \geq 0$, then we return to the discussion of Case (I), and conclude that the iterative sequence converges to $p(\boldsymbol{U})$.
(III) If $p_0 \in (p_{in}, +\infty)$, similar to the discussion in Case (b)(III), we can use proof by contradiction to prove that there exists an iterative value p_n belongs to the interval $(-\infty, 0)$, $[0, p(\boldsymbol{U})]$, or $(p(\boldsymbol{U}), p_{in})$. If $p_n \in (-\infty, 0)$, then the convergence cannot be guaranteed. If $p_n \in [0, p(\boldsymbol{U})]$, then we return to the discussion of Case (c)(I) and conclude that the iterative sequence converges to $p(\boldsymbol{U})$. If $p_n \in (p(\boldsymbol{U}), p_{in}]$, then we return to the discussion of Case (c)(II): the iteration sequence either converges to $p(\boldsymbol{U})$, or a negative value appears in the iterative sequence so that the convergence cannot be guaranteed.
In summary, in Case (c), the iteration sequence either remains non-negative and converges to $p(\boldsymbol{U})$, or a negative number appears in the iterative sequence so that the convergence cannot be guaranteed.

The proof is completed. □

As a direct consequence of Theorem 2.7, we have the following conclusion.

Theorem 2.8. *The NR method with $p_0 = 0$ for solving the equation $\psi(p) = 0$ is always convergent and PCP.*

Theorem 2.9. *If*

$$D < \frac{E^2 - |\boldsymbol{m}|^2}{E}, \tag{2.19}$$

then the NR method for solving $\psi(p) = 0$ with any initial guess $p_0 \geq 0$ is always PCP and convergent.

Proof. First, we prove the PCP property, namely, show that the iteration sequence $\{p_n\}_{n\geq 1}$ are always positive. Assume that $p_n \geq 0$, then it suffices to prove $p_{n+1} = p_n - \frac{\psi(p_n)}{\psi'(p_n)} > 0$. Recall that $\psi'(p) > 0$ for all $p \in [0, +\infty)$. Note that $p_n - \frac{\psi(p_n)}{\psi'(p_n)} > 0$ is equivalent to

$$\begin{aligned}\psi(p_n) &= (E + p_n)\left(\frac{p_n}{\gamma - 1} - E\right) + |\boldsymbol{m}|^2 + D\sqrt{(E + p_n)^2 - |\boldsymbol{m}|^2} \\ &< \left(\frac{2p_n + (2-\gamma)E}{\gamma - 1} + \frac{D(E + p_n)}{\sqrt{(E + p_n)^2 - |\boldsymbol{m}|^2}}\right)p_n = p_n\psi'(p_n),\end{aligned}$$

which is equivalent to

$$D(\gamma - 1) < \frac{p_n^2 - |\boldsymbol{m}|^2(\gamma - 1) + E^2(\gamma - 1)}{\frac{E^2 - |\boldsymbol{m}|^2 + Ep_n}{\sqrt{(E+p_n)^2 - |\boldsymbol{m}|^2}}}. \tag{2.20}$$

Noting that $p_n^2-|\boldsymbol{m}|^2(\gamma-1)+E^2(\gamma-1)\geq(\gamma-1)(E^2-|\boldsymbol{m}|^2)>0$, $\frac{E^2-|\boldsymbol{m}|^2+Ep_n}{\sqrt{(E+p_n)^2-|\boldsymbol{m}|^2}}>0$, and

$$\frac{\partial}{\partial p}\left(\frac{E^2-|\boldsymbol{m}|^2+Ep}{\sqrt{(E+p)^2-|\boldsymbol{m}|^2}}\right)=\frac{|\boldsymbol{m}|^2 p}{\left((E+p)^2-|\boldsymbol{m}|^2\right)^{\frac{3}{2}}}>0,\quad \forall p\in(0,+\infty),$$

we obtain

$$\frac{p_n^2-|\boldsymbol{m}|^2(\gamma-1)+E^2(\gamma-1)}{\frac{E^2-|\boldsymbol{m}|^2+Ep_n}{\sqrt{(E+p_n)^2-|\boldsymbol{m}|^2}}}>\frac{(\gamma-1)(E^2-|\boldsymbol{m}|^2)}{\lim\limits_{p\to+\infty}\left(\frac{E^2-|\boldsymbol{m}|^2+Ep}{\sqrt{(E+p)^2-|\boldsymbol{m}|^2}}\right)}=(\gamma-1)\frac{E^2-|\boldsymbol{m}|^2}{E}.$$

Therefore, if $D<\frac{E^2-|\boldsymbol{m}|^2}{E}$, then

$$D(\gamma-1)<(\gamma-1)\frac{E^2-|\boldsymbol{m}|^2}{E}<\frac{p_n^2-|\boldsymbol{m}|^2(\gamma-1)+E^2(\gamma-1)}{\frac{E^2-|\boldsymbol{m}|^2+Ep_n}{\sqrt{(E+p_n)^2-|\boldsymbol{m}|^2}}},$$

which implies (2.20). Hence, $p_n\geq 0$ implies $p_{n+1}>0$. By induction, we know that the iteration sequence $\{p_n\}_{n\geq 1}$ are always positive. Thanks to Theorem 2.7, we know that the NR iteration converges. The proof is completed. □

Inspired by Theorems 2.8 and 2.9, we design the following algorithm.

Algorithm 2.3 *(NR-II).* The NR iteration reads

$$p_{n+1}=p_n-\frac{\psi(p_n)}{\psi'(p_n)}$$

with p_0 given by

$$p_0=\begin{cases}0, & \text{if } D\geq\frac{E^2-|\boldsymbol{m}|^2}{E},\\ p_c^{\mu}, & \text{otherwise,}\end{cases}\tag{2.21}$$

where the expression of p_c^{μ} is given in (2.16) and the expression of $\psi(p)$ is given in (2.14).

Theorem 2.10. *Algorithm 2.3 is always PCP and convergent. Furthermore, it has a quadratic convergence.*

Proof. If $D\geq\frac{E^2-|\boldsymbol{m}|^2}{E}$, then $p_0=0$, so that Algorithm 2.3 is PCP and convergent, according to Theorem 2.8. If $D<\frac{E^2-|\boldsymbol{m}|^2}{E}$, then Theorem 2.9 implies that Algorithm 2.3 is PCP and convergent. Recalling that $\psi'(p)>0$ for all $p\in[0,+\infty)$, we know that $p(\boldsymbol{U})$ is not a repeated root of $\psi(p)=0$. Therefore, the convergence rate of this NR method is quadratic. □

2.4. Hybrid NR method: hybrid PCP convergent NR iteration

We have proposed two NR methods for recovering primitive variables. Algorithm 2.1 is based on the NR iteration for the polynomial $\phi(p)$. When the polynomial $\phi(p)$ is ill-conditioned, small perturbations in the coefficients c_i $(i=0,1,2,3)$ can lead to significant changes in the root of $\phi(p)=0$, which can result in a reduced accuracy of Algorithm 2.1. Nonetheless, since $\phi(p)$ is a polynomial, the computational cost of evaluating $\phi(p)$ and $\phi'(p)$ in each iteration of Algorithm 2.1 is relatively low. Moreover, the monotonic convergence of Algorithm 2.1 allows for a simpler and more effective stopping criterion, making it computationally faster (as we will show in Section 5). In contrast, Algorithm 2.3 requires taking a square root at each iteration to evaluate $\psi(p)$ and $\psi'(p)$, which leads to slower computation speed, but it does not suffer from the ill-conditioned issue and always provides higher accuracy.

In this subsection, we will propose a hybrid approach that switches to Algorithm 2.3 when $\phi(p)$ is ill-conditioned and to Algorithm 2.1 when $\phi(p)$ is not ill-conditioned, to obtain a NR method that achieves both fast convergence and high accuracy.

The first problem is, how to detect ill condition of polynomials (2.7) efficiently and conveniently. In fact, a polynomial might be ill-conditioned when cluster roots occur [8]. In the following, we observe and prove that when D or $(\gamma-1)$ is very small, p_a^{μ} will be very close to $p(\boldsymbol{U})$, which results in the polynomial $\phi(p)$ becoming ill-conditioned.

Lemma 2.7. *For fixed $\gamma,\boldsymbol{m}$, and E, we have* $\lim\limits_{D\to 0^+}|p_a^{\mu}-p(\boldsymbol{U})|=0$.

Proof. Note that $h_1(p)=h_3^2(p)$ is independent of D, and

$$\lim_{D\to 0^+}h_2(p)=\lim_{D\to 0^+}D^2\left((E+p)^2-|\boldsymbol{m}|^2\right)=0.$$

Since $\phi(p) = (\gamma-1)^2\left[h_3^2(p) - h_2(p)\right]$, when D approaches zero, p_a^μ and $p(\boldsymbol{U})$ approaches p_b^μ, which is the unique positive root of $h_3(p)$ as shown in the proof of Lemma 2.2. □

In practical computation, the ratio $\frac{h_2(0)}{h_1(0)} = \frac{D^2}{E^2-|\boldsymbol{m}|^2}$ can be used to estimate whether D is small enough to result in clustered roots. Note that $E > \sqrt{D^2+|\boldsymbol{m}|^2}$, which implies $\frac{D^2}{E^2-|\boldsymbol{m}|^2} < 1$.

Lemma 2.8. *For fixed conservative quantities $D, \boldsymbol{m}$, and E, we have $\lim_{\gamma\to 1^+} p(\boldsymbol{U}) = \lim_{\gamma\to 1^+} p_a^\mu = 0$, and $\lim_{\gamma\to 1^+} |p_a^\mu - p(\boldsymbol{U})| = 0$.*

Proof. Consider $h_4(p) := c_1 + c_2 p + c_3 p^2$, where $c_1 < 0$ and $c_3 > 0$ when $\gamma < 2$ according to Lemma 2.1. If $\gamma < 2$, then $h_4(p) > 0$ when $p > \frac{-c_2+\sqrt{c_2^2-4c_1c_3}}{2c_3}$. When $\gamma < 1.2 < 3-\sqrt{3}$, we have

$$c_2 = E^2(\gamma^2-6\gamma+6) + 2|\boldsymbol{m}|^2(\gamma-1) - D^2(\gamma-1)^2 > 0.24E^2 - 0.04D^2 > 0.2E^2 > 0,$$

which yields

$$\frac{-c_2+\sqrt{c_2^2-4c_1c_3}}{2c_3} = \frac{-2c_1}{c_2+\sqrt{c_2^2-4c_1c_3}} < -\frac{c_1}{c_2} < -\frac{c_1}{0.2E^2}.$$

Thus, if $\gamma < 1.2$ and $p > -\frac{c_1}{0.2E^2}$, then $h_4(p) > 0$ and $\phi(p) = h_4(p)p + c_0 + p^4 > 0$, where we have used $c_0 > 0$ proved in Lemma 2.1. Therefore,

$$0 < p(\boldsymbol{U}) < p_a^\mu < -\frac{c_1}{0.2E^2} = 10(\gamma-1)\frac{E(2-\gamma)(E^2-|\boldsymbol{m}|^2) + ED^2(\gamma-1)}{E^2},$$

which implies $\lim_{\gamma\to 1^+} p(\boldsymbol{U}) = \lim_{\gamma\to 1^+} p_a^\mu = 0$. It follows that $\lim_{\gamma\to 1^+} |p_a^\mu - p(\boldsymbol{U})| = 0$. The proof is completed. □

By utilizing Lemmas 2.7 and 2.8, we can effectively detect the ill-conditioned problem: the polynomial $\phi(p)$ may be ill-conditioned if $\gamma < 1+\epsilon_1$ or $\frac{D^2}{E^2-|\boldsymbol{m}|^2} < \epsilon_2$, where $\epsilon_1 \in (0,1)$ and $\epsilon_2 \in (0,1)$ are two small positive numbers.

Remark 2.3. When p_a^μ and $p(\mathbf{U})$ are in close proximity, the polynomial $\phi(p)$ tends to become ill-conditioned, thereby compromising the accuracy of Algorithm 2.1. On the other hand, Algorithm 2.3 demonstrates high accuracy and rapid convergence when p_a^μ and $p(\mathbf{U})$ are in close proximity. This is due to the fact that p_c^μ is positioned between $p(U)$ and p_a^μ, resulting in the initial value p_0 for Algorithm 2.3 that is often very close to the actual pressure $p(\boldsymbol{U})$ in such cases. In other words, Algorithm 2.3 exactly compensates for the limitations of Algorithm 2.1.

Based on the aforementioned discussions, we propose the following hybrid method.

Algorithm 2.4 *(Hybrid NR).*

$$\begin{cases} \text{Adopt Algorithm 2.1}, & \text{if } \gamma \geq 1+\epsilon_1 \text{ and } \frac{D^2}{E^2-|\boldsymbol{m}|^2} \geq \epsilon_2. \\ \text{Adopt Algorithm 2.3}, & \text{otherwise}, \end{cases} \tag{2.22}$$

where $\epsilon_1 \in (0,1)$ and $\epsilon_2 \in (0,1)$ are two small positive numbers. In this paper, we set $\epsilon_1 = 0.01$ and $\epsilon_2 = 10^{-4}$.

Remark 2.4. Since both the NR-I and NR-II methods are always PCP and convergent with quadratic convergence rate, the above hybrid NR method is also PCP and convergent with quadratic convergence rate.

If $\gamma \geq 1+\epsilon_1$ and $\frac{D^2}{E^2-|\boldsymbol{m}|^2} \geq \epsilon_2$, Algorithm 2.4 utilizes Algorithm 2.1, which exhibits fast convergence and high accuracy with low computational cost since the polynomial $\phi(x)$ is not ill-conditioned in this scenario. In contrast, when $\gamma < 1+\epsilon_1$ or $\frac{D^2}{E^2-|\boldsymbol{m}|^2} < \epsilon_2$, Algorithm 2.4 switches to Algorithm 2.3, and in this case, the initial value p_0 of Algorithm 2.3 (defined in (2.21)) is typically in close proximity to the physical pressure $p(\boldsymbol{U})$ (as discussed in Remark 2.3), thus ensuring both fast convergence and high accuracy. Overall, the hybrid NR method effectively combines the advantages of both NR-I and NR-II methods and circumvents their individual limitations, thereby achieving both high accuracy and fast convergence, as confirmed by numerical tests in Section 5.

3. One-dimensional PCP HWENO scheme

In this section, we present the PCP finite volume HWENO scheme for the 1D special RHD equations

$$\frac{\partial \boldsymbol{U}}{\partial t}+\frac{\partial \boldsymbol{F}(\boldsymbol{U})}{\partial x}=\boldsymbol{0}, \tag{3.1}$$

where

$$\boldsymbol{U}=\left(D, m_1, E\right)^{\top}, \quad \boldsymbol{F}=\left(Dv_i, m_1 v_1+p, m_1\right)^{\top}.$$

Divide the computational domain into N uniform cells $I_i=\left[x_{i-1/2}, x_{i+1/2}\right]$, $1\le i\le N$, with the cell center $x_i=\frac{1}{2}(x_{i-1/2}+x_{i+1/2})$. Let Δx denote the mesh size $x_{i+1/2}-x_{i-1/2}$.

3.1. 1D PCP finite volume HWENO scheme

In this subsection, we give the 1D finite volume PCP HWENO scheme. For the notational convenience, we denote $x_{i+a}=x_i+a\Delta x$, where $a\in[-\frac{1}{2},\frac{1}{2}]$ is a real number.

The semi-discrete finite volume HWENO scheme for the RHD equations (3.1) is given by

$$\frac{\mathrm{d}\overline{\boldsymbol{U}}_i(t)}{\mathrm{d}t}=-\frac{\hat{\boldsymbol{F}}_{i+1/2}-\hat{\boldsymbol{F}}_{i-1/2}}{\Delta x}=:\mathcal{L}_U(\boldsymbol{U}_h(t),i), \tag{3.2}$$

$$\frac{\mathrm{d}\overline{\boldsymbol{V}}_i(t)}{\mathrm{d}t}=-\frac{\hat{\boldsymbol{F}}_{i+1/2}+\hat{\boldsymbol{F}}_{i-1/2}}{2\Delta x}+\frac{\boldsymbol{H}_i}{\Delta x}=:\mathcal{L}_V(\boldsymbol{U}_h(t),i), \tag{3.3}$$

where

$$\overline{\boldsymbol{U}}_i(t)\approx\frac{1}{\Delta x}\int_{I_i}\boldsymbol{U}(x,t)\mathrm{d}x, \quad \overline{\boldsymbol{V}}_i(t)\approx\frac{1}{\Delta x}\int_{I_i}\boldsymbol{U}(x,t)\frac{x-x_i}{\Delta x}\mathrm{d}x$$

denote the approximations to the zeroth and first order moments in I_i, respectively,

$$\boldsymbol{U}_h(t)=\{\overline{\boldsymbol{U}}_i(t)\}_{i=1}^N\cup\{\overline{\boldsymbol{V}}_i(t)\}_{i=1}^N$$

denotes the set that contains all cells' zeroth and first order moments, and

$$\boldsymbol{H}_i=\sum_{l=1}^{4}\hat{\omega}_\ell\boldsymbol{F}(\boldsymbol{U}_{i\oplus a_\ell})\approx\frac{1}{\Delta x}\int_{I_i}\boldsymbol{F}(\boldsymbol{U})\,\mathrm{d}x$$

with $\boldsymbol{U}_{i\oplus a_\ell}$ denoting the approximation to $\boldsymbol{U}(x_{i+a_\ell},t)$ within the cell I_i. Here the four-point Gauss–Lobatto quadrature is used with the quadrature weights and nodes given by

$$\hat{\omega}_1=\hat{\omega}_4=\frac{1}{12}, \qquad \hat{\omega}_2=\hat{\omega}_3=\frac{5}{12},$$

$$x_{i+a_\ell}=x_i+a_\ell\Delta x, \qquad \{a_\ell\}_{\ell=1}^4=\left\{-\frac{1}{2}, -\frac{\sqrt{5}}{10}, \frac{\sqrt{5}}{10}, \frac{1}{2}\right\}.$$

In (3.2)–(3.3), $\hat{\boldsymbol{F}}_{i+1/2}$ denotes the numerical flux at the cell interface point $x_{i+1/2}$. In this paper, we employ the Lax–Friedrichs numerical flux[1]

$$\hat{\boldsymbol{F}}_{i+1/2}=\frac{1}{2}\left(\boldsymbol{F}_1\left(\boldsymbol{U}_{i\oplus\frac{1}{2}}\right)+\boldsymbol{F}_1\left(\boldsymbol{U}_{(i+1)\oplus\left(-\frac{1}{2}\right)}\right)-\alpha\left(\boldsymbol{U}_{(i+1)\oplus\left(-\frac{1}{2}\right)}-\boldsymbol{U}_{i\oplus\frac{1}{2}}\right)\right), \tag{3.4}$$

which will be useful for achieving the PCP property of our HWENO scheme. Here α is defined by

$$\alpha=\max_i\max\left\{\rho_1\left(\boldsymbol{U}_{i\oplus\frac{1}{2}}\right), \rho_1\left(\boldsymbol{U}_{(i+1)\oplus\left(-\frac{1}{2}\right)}\right)\right\},$$

where $\rho_1(\boldsymbol{U})$ denotes the spectral radius of the Jacobian matrix $\frac{\partial\boldsymbol{F}_1(\boldsymbol{U})}{\partial\boldsymbol{U}}$.

Remark 3.1. It should be pointed out that the symbol "$\oplus$" in the subscript of $\boldsymbol{U}_{i\oplus a}$ with $a\in[-\frac{1}{2},\frac{1}{2}]$ is not a standard addition operation, but rather a symbol representing the position relative to the cell center x_i. For example, $\boldsymbol{U}_{i\oplus\frac{1}{2}}$ represents the approximate

[1] Another option is the Harten-Lax-van Leer (HLL) numerical flux, whose PCP property was proved in [4].

value at the point $x_i + \frac{1}{2}\Delta x$ computed within the cell I_i, while $\boldsymbol{U}_{(i+1)\oplus\left(-\frac{1}{2}\right)}$ stands for the approximate value at the same point $x_{i+1} - \frac{1}{2}\Delta x$ but computed within the cell I_{i+1}.

To compute $\mathcal{L}_U(\boldsymbol{U}_h(t), i)$ and $\mathcal{L}_V(\boldsymbol{U}_h(t), i)$ in (3.2)–(3.3), one needs to reconstruct the point values $\{\boldsymbol{U}_{i\oplus a_\ell}\}_{\ell=1}^4$ by using $\{\overline{\boldsymbol{U}}_i\}$ and $\{\overline{\boldsymbol{V}}_i\}$. To facilitate the subsequent description, we first introduce two reconstruction operators $\boldsymbol{M}_L(\cdot,\cdot,\cdot)$ and $\boldsymbol{M}_H(\cdot,\cdot,\cdot)$, before giving the detailed spatial reconstruction procedures of our 1D PCP finite volume HWENO scheme.

3.1.1. Linear reconstruction operator $\boldsymbol{M}_L$

Let us reconstruct a quintic polynomial $P_0(x) = \sum_{l=0}^{5} a_l^0 \left(\frac{x-x_i}{\Delta x}\right)^l$ satisfying

$$\frac{1}{\Delta x}\int_{I_k} P_0(x)\,\mathrm{d}x = u_k,\ k = i, i\pm 1, \tag{3.5}$$

$$\frac{1}{\Delta x}\int_{I_k} P_0(x)\,\frac{x-x_i}{\Delta x}\mathrm{d}x = v_k,\ k = i, i\pm 1, \tag{3.6}$$

where u_k and v_k are given real numbers. The six equations (3.5)–(3.6) form a linear algebraic system for the unknowns $\{a_l^0\}_{l=0}^5$. Solving the system gives the expressions of a_l^0, which are the linear combinations of u_i, v_i, $u_{i\pm1}$, $v_{i\pm1}$ given by

$$\begin{cases} a_0^0 = -\frac{43}{384}u_{i-1} + \frac{235}{192}u_i - \frac{43}{384}u_{i+1} - \frac{27}{64}v_{i-1} + \frac{27}{64}v_{i+1}, \\ a_1^0 = \frac{167}{576}u_{i-1} - \frac{167}{576}u_{i+1} + \frac{281}{288}v_{i-1} + \frac{2449}{144}v_i + \frac{281}{288}v_{i+1}, \\ a_2^0 = \frac{23}{16}u_{i-1} - \frac{23}{8}u_i + \frac{23}{16}u_{i+1} + \frac{45}{8}v_{i-1} - \frac{45}{8}v_{i+1}, \\ a_3^0 = -\frac{455}{216}u_{i-1} + \frac{455}{216}u_{i+1} - \frac{785}{108}v_{i-1} - \frac{1945}{54}v_i - \frac{785}{108}v_{i+1}, \\ a_4^0 = -\frac{5}{8}u_{i-1} + \frac{4}{5}u_i - \frac{5}{8}u_{i+1} - \frac{15}{4}v_{i-1} + \frac{15}{4}v_{i+1}, \\ a_5^0 = \frac{35}{36}u_{i-1} - \frac{35}{36}u_{i+1} + \frac{77}{18}v_{i-1} + \frac{133}{9}v_i + \frac{77}{18}v_{i+1}. \end{cases}$$

Define $\eta = \frac{x-x_i}{\Delta x}$ and the operator

$$M_L([u_{i-1}\ u_i\ u_{i+1}], [v_{i-1}\ v_i\ v_{i+1}], \eta) := P_0(x(\eta)) = \sum_{l=0}^{5} a_l^0 \eta^l,$$

which is a mapping from $\mathbb{R}^{1\times3}\times\mathbb{R}^{1\times3}\times\mathbb{R}$ to $\mathbb{R}$. Using this operator, it is easy to compute the value of $P_0(x) =$ at $x_{i+\eta}$ with $P_0(x_{i+\eta}) = M_L([u_{i-1}\ u_i\ u_{i+1}], [v_{i-1}\ v_i\ v_{i+1}], \eta)$. For example,

$$P_0(x_{i+1/2}) = M_L([u_{i-1}\ u_i\ u_{i+1}], [v_{i-1}\ v_i\ v_{i+1}], \frac{1}{2}) = \frac{13}{108}u_{i-1} + \frac{7}{12}u_i + \frac{8}{27}u_{i+1} + \frac{25}{54}v_{i-1} + \frac{241}{54}v_i - \frac{28}{27}v_{i+1},$$

which is independent of the cell size Δx and the cell center x_i.

The operator M_L represents the reconstruction mapping for the scalar equation. In order to extend the reconstruction to the 1D RHD equations, we generalize the operator to vector cases component-wisely as follows

$$\boldsymbol{M}_L([\boldsymbol{U}_1\ \boldsymbol{U}_2\ \boldsymbol{U}_3], [\boldsymbol{V}_1\ \boldsymbol{V}_2\ \boldsymbol{V}_3], \eta) := \begin{pmatrix} M_L([U_1^{(1)}\ U_2^{(1)}\ U_3^{(1)}], [V_1^{(1)}\ V_2^{(1)}\ V_3^{(1)}], \eta) \\ M_L([U_1^{(2)}\ U_2^{(2)}\ U_3^{(2)}], [V_1^{(2)}\ V_2^{(2)}\ V_3^{(2)}], \eta) \\ M_L([U_1^{(3)}\ U_2^{(3)}\ U_3^{(3)}], [V_1^{(3)}\ V_2^{(3)}\ V_3^{(3)}], \eta) \end{pmatrix},$$

where $\boldsymbol{M}_L$ is the reconstruction operator from $\mathbb{R}^{3\times3}\times\mathbb{R}^{3\times3}\times\mathbb{R}$ to $\mathbb{R}^{3\times1}$. It is worth pointing out that $\boldsymbol{M}_L(\cdot,\cdot,\eta)$ is also a linear mapping for a fixed η.

3.1.2. HWENO reconstruction operator $\boldsymbol{M}_H$

Consider two quadratic polynomials $P_1(x) = \sum_{l=0}^{2} a_l^1 \left(\frac{x-x_i}{\Delta x}\right)^l$ and $P_2(x) = \sum_{l=0}^{2} a_l^2 \left(\frac{x-x_i}{\Delta x}\right)^l$ satisfying

$$\frac{1}{\Delta x}\int_{I_i} P_1(x)\,\frac{x-x_i}{\Delta x}\mathrm{d}x = v_i, \quad \frac{1}{\Delta x}\int_{I_k} P_1(x)\,\mathrm{d}x = u_k, \quad k = i, i-1,$$

$$\frac{1}{\Delta x}\int_{I_i} P_2(x)\,\frac{x-x_i}{\Delta x}\mathrm{d}x = v_i, \quad \frac{1}{\Delta x}\int_{I_k} P_2(x)\,\mathrm{d}x = u_k, \quad k = i, i+1.$$

Similarly, we can obtain the expressions of a_l^1 and a_l^2, which are also linear combinations of u_i, v_i, $u_{i\pm1}$, $v_{i\pm1}$, given by

$$\begin{cases} a_0^1 = -\frac{1}{12}u_{i-1} + \frac{13}{12}u_i - v_i, \\ a_1^1 = 12v_i, \\ a_2^1 = u_{i-1} - u_i + 12v_i, \\ a_0^2 = \frac{13}{12}u_i - \frac{1}{12}u_{i+1} + v_i, \\ a_1^2 = 12v_i, \\ a_2^2 = -u_i + u_{i+1} - 12v_i. \end{cases}$$

Next, in order to measure the smoothness of the polynomial $P_n(x)$ in the cell I_i, we calculate the smooth indicators, with the same definition as in [62],

$$\beta_n = \sum_{\alpha=1}^{r}\int_{I_i} \Delta x^{2\alpha-1}\left(\frac{\mathrm{d}^\alpha P_n(x)}{\mathrm{d}x^\alpha}\right)^2 \mathrm{d}x, \ n = 0,1,2, \tag{3.7}$$

where r is the degree of the polynomials $P_n(x)$. The expressions of β_n are

$$\begin{cases} \beta_0 = \left(\frac{19}{108}u_{i-1} - \frac{19}{108}u_{i+1} + \frac{31}{54}v_{i-1} - \frac{241}{27}v_i + \frac{31}{54}v_{i+1}\right)^2 + \left(\frac{9}{4}u_{i-1} - \frac{9}{2}u_i + \frac{9}{4}u_{i+1} + \right. \\ \quad \left. \frac{15}{2}v_{i-1} - \frac{15}{2}v_{i+1}\right)^2 + \left(\frac{70}{9}u_{i-1} - \frac{70}{9}u_{i+1} + \frac{200}{9}v_{i-1} + \frac{1280}{9}v_i + \frac{200}{9}v_{i+1}\right)^2 + \\ \quad \frac{1}{12}\left(\frac{5}{2}u_{i-1} - 5u_i + \frac{5}{2}u_{i+1} + 9v_{i-1} - 9v_{i+1}\right)^2 + \frac{1}{12}\left(\frac{175}{18}u_{i-1} - \frac{175}{18}u_{i+1} + \frac{277}{9}v_{i-1} + \right. \\ \quad \left. \frac{1546}{9}v_i + \frac{277}{9}v_{i+1}\right)^2 + \frac{1}{180}\left(\frac{95}{18}u_{i-1} - \frac{95}{18}u_{i+1} + \frac{155}{9}v_{i-1} + \frac{830}{9}v_i + \frac{155}{9}v_{i+1}\right)^2 + \\ \quad \frac{109341}{175}\left(\frac{5}{8}u_{i-1} - \frac{5}{4}u_i + \frac{5}{8}u_{i+1} + \frac{15}{4}v_{i-1} - \frac{15}{4}v_{i+1}\right)^2 + \frac{27553933}{1764}\left(\frac{35}{36}u_{i-1} - \frac{35}{36}u_{i+1} + \right. \\ \quad \left. \frac{77}{18}v_{i-1} + \frac{133}{9}v_i + \frac{77}{18}v_{i+1}\right)^2, \\ \beta_1 = 144v_i^2 + \frac{13}{3}(u_{i-1} - u_i + 12v_i)^2, \\ \beta_2 = 144v_i^2 + \frac{13}{3}(u_i - u_{i+1} + 12v_i)^2. \end{cases}$$

Then the HWENO reconstruction polynomial is defined by

$$P_H(x) = \omega_0\left(\frac{1}{\gamma_0}P_0(x) - \sum_{n=1}^{2}\frac{\gamma_n}{\gamma_0}P_n(x)\right) + \sum_{n=1}^{2}\omega_n P_n(x),$$

where the nonlinear weights

$$\omega_n = \frac{\bar{\omega}_n}{\sum_{k=0}^{2}\bar{\omega}_k} \quad \text{with} \quad \bar{\omega}_n = \gamma_n\left(1 + \frac{\tau^2}{\beta_n^2 + \epsilon}\right), \ n = 0,1,2, \tag{3.8}$$

$\tau := \frac{|\beta_0-\beta_1|+|\beta_0-\beta_2|}{2}$, and ϵ is a tiny positive number to avoid the denominator being zero. These nonlinear weights possess a "scale-invariant" property, which means that the nonlinear weights $\{\omega_n\}$ remain unchanged when $\{u_i, v_i, u_{i\pm1}, v_{i\pm1}\}$ are replaced by $\{\lambda u_i, \lambda v_i, \lambda u_{i\pm1}, \lambda v_{i\pm1}\}$ for any $\lambda \neq 0$.

Let $\eta := \frac{x-x_i}{\Delta x}$. Define the operator

$$\begin{aligned} M_H([u_{i-1}\ u_i\ u_{i+1}],[v_{i-1}\ v_i\ v_{i+1}],\eta) &:= P_H(x(\eta)) \\ &= \omega_0\left(\frac{1}{\gamma_0}\sum_{l=0}^{5}a_l^0\eta^l - \sum_{n=1}^{2}\frac{\gamma_n}{\gamma_0}\sum_{l=0}^{5}a_l^n\eta^l\right) + \sum_{n=1}^{2}\omega_n\sum_{l=0}^{5}a_l^n\eta^l, \end{aligned}$$

which is a mapping from $\mathbb{R}^{1\times3}\times\mathbb{R}^{1\times3}\times\mathbb{R}$ to $\mathbb{R}$. It is easy to compute the value of $P_H(x)$ at $x_{i+\eta}$ with $P_H(x_{i+\eta}) = M_H([u_{i-1}\ u_i\ u_{i+1}],[v_{i-1}\ v_i\ v_{i+1}],\eta)$. We can generalize the scalar HWENO reconstruction operator M_H to the vector cases in a component by component manner:

$$\boldsymbol{M}_H([\boldsymbol{U}_1\ \boldsymbol{U}_2\ \boldsymbol{U}_3],[\boldsymbol{V}_1\ \boldsymbol{V}_2\ \boldsymbol{V}_3],\eta) := \begin{pmatrix} M_H([U_1^{(1)}\ U_2^{(1)}\ U_3^{(1)}],[V_1^{(1)}\ V_2^{(1)}\ V_3^{(1)}],\eta) \\ M_H([U_1^{(2)}\ U_2^{(2)}\ U_3^{(2)}],[V_1^{(2)}\ V_2^{(2)}\ V_3^{(2)}],\eta) \\ M_H([U_1^{(3)}\ U_2^{(3)}\ U_3^{(3)}],[V_1^{(3)}\ V_2^{(3)}\ V_3^{(3)}],\eta) \end{pmatrix},$$

where $U_i^{(\ell)}$ is the ℓth component of $\boldsymbol{U}_i$, $V_i^{(\ell)}$ is the ℓth component of $\boldsymbol{V}_i$. Different from $\boldsymbol{M}_L$, the operator $\boldsymbol{M}_H(\cdot,\cdot,\eta)$ is a nonlinear mapping for a fixed η.

Remark 3.2. When solving the relativistic hydrodynamics (RHD) equations, the wide range of variable scales arising from relativistic effects in the ultra-relativistic regime can significantly impact the effectiveness of shock capturing. As recently demonstrated in [4], using scale-invariant nonlinear weights can effectively suppress oscillations for simulating multiscale RHD problems. Based on numerical tests, we have also observed that adopting scale-invariant nonlinear weights (3.8) is vital not only for the RHD equations but also for controlling spurious oscillations for multi-scale problems of other hyperbolic equations, such as the Euler equations.

3.1.3. Detailed PCP HWENO reconstruction procedure

We are now in a position to present the detailed PCP HWENO reconstruction procedure of our 1D HWENO scheme.

Step 1. Use the KXRCF indicator [20] to identify the troubled cells where the solution may be discontinuous. Then modify the first-order moment in the troubled cells by using the HWENO limiter given in [62]. The adoption of the KXRCF indicator in HWENO schemes is motivated by [62], which demonstrated its efficacy in reducing computational costs. We observe that the nonlinear weights in the HWENO limiter are also necessary to be scale-invariant, thus we modify the nonlinear weights in the HWENO limiter [62] to $\omega_n^l = \frac{\bar{\omega}_n^l}{\sum_{k=0}^{2}\bar{\omega}_k^l}$ with $\bar{\omega}_n^l = \gamma_n\left(1+\frac{\tau_l^2\Delta x}{(\beta_n^l)^2+\epsilon}\right), n=0,1,2$.

Step 2. Reconstruct the point values of the solution at the four Gauss–Lobatto points.

- If cell I_i is not a troubled cell, employ the linear reconstruction:

$$\boldsymbol{U}^*_{i\oplus a_\ell} = \boldsymbol{M}_L\left(\left[\overline{\boldsymbol{U}}_{i-1}\ \overline{\boldsymbol{U}}_i\ \overline{\boldsymbol{U}}_{i+1}\right],\left[\overline{\boldsymbol{V}}_{i-1}\ \overline{\boldsymbol{V}}_i\ \overline{\boldsymbol{V}}_{i+1}\right],a_\ell\right),\quad \ell\in\{1,2,3,4\}.$$

- If I_i is a troubled cell, use the HWENO reconstruction.

  (i) Perform HWENO reconstruction in a component-by-component fashion for the second and third Gauss-Lobatto points:

$$\boldsymbol{U}^*_{i\oplus a_\ell} = \boldsymbol{M}_H\left(\left[\overline{\boldsymbol{U}}_{i-1}\ \overline{\boldsymbol{U}}_i\ \overline{\boldsymbol{U}}_{i+1}\right],\left[\overline{\boldsymbol{V}}_{i-1}\ \overline{\boldsymbol{V}}_i\ \overline{\boldsymbol{V}}_{i+1}\right],a_\ell\right),\quad \ell\in\{2,3\}.$$

  (ii) Perform HWENO reconstruction based on characteristic decomposition for the cell interface points:

$$\boldsymbol{U}^*_{i\oplus\left(\pm\frac{1}{2}\right)} = \boldsymbol{R}_{i\pm\frac{1}{2}}\boldsymbol{M}_H\left(\boldsymbol{R}^{-1}_{i\pm\frac{1}{2}}\left[\overline{\boldsymbol{U}}_{i-1}\ \overline{\boldsymbol{U}}_i\ \overline{\boldsymbol{U}}_{i+1}\right],\boldsymbol{R}^{-1}_{i\pm\frac{1}{2}}\left[\overline{\boldsymbol{V}}_{i-1}\ \overline{\boldsymbol{V}}_i\ \overline{\boldsymbol{V}}_{i+1}\right],\pm\frac{1}{2}\right),$$

  where $\boldsymbol{R}^{-1}_{i+1/2}$ and $\boldsymbol{R}_{i+1/2}$ are taken as left and right eigenvector matrices of the Roe matrix [9] at $x_{i+1/2}$. The eigenvector matrices and fluxes are computed by using the primitive variables, which are recovered by using the proposed NR methods.

Step 3. Perform the PCP limiter on the reconstructed point values $\{\boldsymbol{U}^*_{i\oplus a_\ell}\}_{\ell=1}^4$ as follows. Define $\boldsymbol{U}^*_{i\oplus a_\ell} =: (D^*_{i\oplus a_\ell},(m_1)^*_{i\oplus a_\ell},E^*_{i\oplus a_\ell})^\top$ and the first component of $\overline{\boldsymbol{U}}_i$ as $\overline{D}_i$.

- Modify the mass density to enforce its positivity via

$$\widetilde{D}_{i\oplus a_\ell} = \theta_D(D^*_{i\oplus a_\ell}-\overline{D}_i)+\overline{D}_i \quad\text{with}\quad \theta_D = \min\left\{\left|\frac{\overline{D}_i-\epsilon_D}{\overline{D}_i-D_{\min}}\right|,1\right\},$$

$$D_{\min} = \min\left\{\min_\ell\{D^*_{i\oplus a_\ell}\},\frac{\overline{D}_i-\hat{\omega}_1\boldsymbol{U}^*_{i\oplus\left(-\frac{1}{2}\right)}-\hat{\omega}_4\boldsymbol{U}^*_{i\oplus\frac{1}{2}}}{1-2\hat{\omega}_1}\right\},$$

  where $\epsilon_D = \min_i\left\{10^{-13},\overline{D}_i\right\}$ is a small positive number introduced to avoid the effect of round-off errors.

- Define $\widetilde{\boldsymbol{U}}_{i\oplus a_\ell} = (\widetilde{D}_{i\oplus a_\ell},(m_1)^*_{i\oplus a_\ell},E^*_{i\oplus a_\ell})^\top$. Enforce the positivity of $g(\boldsymbol{U})$ by

$$\boldsymbol{U}_{i\oplus a_\ell} = \theta_g(\widetilde{\boldsymbol{U}}_{i\oplus a_\ell}-\overline{\boldsymbol{U}}_i)+\overline{\boldsymbol{U}}_i \quad\text{with}\quad \theta_g = \min\left\{\left|\frac{g(\overline{\boldsymbol{U}}_i)-\epsilon_g}{g(\overline{\boldsymbol{U}}_i)-g_{\min}}\right|,1\right\}, \tag{3.9}$$

$$g_{\min} = \min\left\{\min_\ell g(\widetilde{\boldsymbol{U}}^*_{i\oplus a_\ell}),g\left(\frac{\overline{\boldsymbol{U}}_i-\hat{\omega}_1\widetilde{\boldsymbol{U}}^*_{i\oplus\left(-\frac{1}{2}\right)}-\hat{\omega}_4\widetilde{\boldsymbol{U}}^*_{i\oplus\frac{1}{2}}}{1-2\hat{\omega}_1}\right)\right\},$$

where $\epsilon_g = \min_i \left\{ 10^{-13}, g(\overline{\boldsymbol{U}}_i) \right\}$.

Remark 3.3. The characteristic decomposition is very important for the HWENO reconstruction for the cell interface points. However, the wide range of characteristic variable scales resulting from relativistic effects can lead to large numerical errors in characteristic decomposition, especially in challenging test problems, where this problem is particularly severe in the HWENO scheme. To mitigate this issue, we propose to rescale the characteristic vectors. Let $(\boldsymbol{r}, \boldsymbol{l})$ be a pair of right and left eigenvectors of the Jacobian matrix $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}}$, then $(c\boldsymbol{r}, \boldsymbol{l}/c)$ is also a pair of eigenvectors, where $c \neq 0$. We consider the following two rescaling approaches:

(i) (Unitization) Unitize the left eigenvectors with $c = |\boldsymbol{l}|$. However, this method may result in extremely large values of $|c\boldsymbol{r}|$.
(ii) (Matching) Matching the norms between $c\boldsymbol{r}$ and $\boldsymbol{l}/c$ with $c = \sqrt{\frac{|\boldsymbol{l}|}{|\boldsymbol{r}|}}$, such that $|\boldsymbol{l}|/c = c\,|\boldsymbol{r}|$.

In Section 5, we will present a numerical example to demonstrate the necessity of rescaling eigenvectors and compare the effectiveness of these two approaches. The results show that the "matching" approach is superior to the "unitization" approach.

Remark 3.4. As demonstrated in Section 2, the algorithms recovering primitive variables from $\boldsymbol{U}$ are convergent only when $\boldsymbol{U} \in \mathcal{G}$. Therefore, for the RHD equations, the pointwise PCP property at a given point is necessary whenever the fluxes are computed at that point. This differs from the Euler equations [59]. For this reason, we employ the PCP limiting procedures for the reconstructed values at the inner Gauss–Lobatto points $\boldsymbol{U}_{i+a_\ell}$ $(\ell = 2, 3)$ in Step 3.

3.1.4. Time discretization

To obtain a fully discrete scheme, we use the strong-stability-preserving (SSP) Runge–Kutta methods to further discretize the semi-discrete scheme (3.2)–(3.3) in time. For example, the third-order SSP Runge–Kutta method reads

$$\begin{aligned}
\overline{\boldsymbol{W}}_i^{(1)} &= \overline{\boldsymbol{W}}_i^n + \Delta t \mathcal{L}_W(\boldsymbol{U}_h^n, i), \\
\overline{\boldsymbol{W}}_i^{(2)} &= \frac{3}{4}\overline{\boldsymbol{W}}_i^n + \frac{1}{4}\left(\overline{\boldsymbol{W}}_i^{(1)} + \Delta t \mathcal{L}_W(\boldsymbol{U}_h^{(1)}, i)\right), \\
\overline{\boldsymbol{W}}_i^{n+1} &= \frac{1}{3}\overline{\boldsymbol{W}}_i^n + \frac{2}{3}\left(\overline{\boldsymbol{W}}_i^{(2)} + \Delta t \mathcal{L}_W(\boldsymbol{U}_h^{(2)}, i)\right),
\end{aligned} \tag{3.10}$$

where the symbol "$\boldsymbol{W}$" can be replaced with "$\boldsymbol{U}$" or "$\boldsymbol{V}$".

3.2. Rigorous analysis of PCP property

In this subsection, we present a rigorous analysis of PCP property of the proposed HWENO scheme.
First, we recall the following Lax–Friedrichs splitting property, which was proved for the RHD equations in [50].

Lemma 3.1. *If $\boldsymbol{U} \in \mathcal{G}$, then*

$$\boldsymbol{F}_\ell^{\pm}(\boldsymbol{U}, \alpha) := \boldsymbol{U} \pm \alpha^{-1} \boldsymbol{F}_\ell(\boldsymbol{U}) \in \mathcal{G}$$

for any $\alpha \geq \rho_\ell(\boldsymbol{U})$, $\ell = 1, \ldots, d$.

Similar to [4, Proposition 3.1], we can prove that the PCP limited point values (3.9) satisfy the following property.

Lemma 3.2. *If $\overline{\boldsymbol{U}}_i \in \mathcal{G}$, then the PCP limited point values (3.9) satisfy*

$$\boldsymbol{U}_{i\oplus a_\ell} \in \mathcal{G}, \quad \forall \ell \in \{1, 2, 3, 4\}, \qquad \boldsymbol{\Pi}_i := \frac{1}{1 - 2\hat{\omega}_1}\left(\overline{\boldsymbol{U}}_i - \hat{\omega}_1 \boldsymbol{U}_{i\oplus\left(-\frac{1}{2}\right)} - \hat{\omega}_4 \boldsymbol{U}_{i\oplus\frac{1}{2}}\right) \in \mathcal{G}. \tag{3.11}$$

Based on the above two lemmas, we are now ready to show the PCP property of the proposed HWENO scheme.

Theorem 3.1. *Consider the proposed 1D HWENO scheme (3.2)–(3.3) with the Lax–Friedrichs flux (3.4). If $\overline{\boldsymbol{U}}_i \in \mathcal{G}$ for all i, then the updated cell averages*

$$\overline{\boldsymbol{U}}_i + \Delta t \mathcal{L}_U(\boldsymbol{U}_h(t), i) \in \mathcal{G}, \qquad \forall i \tag{3.12}$$

under the CFL condition

$$\Delta t \leq \frac{\hat{\omega}_1 \Delta x}{\alpha}. \tag{3.13}$$

Proof. Based on (3.11), we have the following convex decomposition for the cell average:

$$\overline{\boldsymbol{U}}_i = (1-2\hat{\omega}_1)\boldsymbol{\Pi}_i + \hat{\omega}_1 \boldsymbol{U}_{i\oplus\left(-\frac{1}{2}\right)} + \hat{\omega}_1 \boldsymbol{U}_{i\oplus\frac{1}{2}}$$

with $\boldsymbol{\Pi}_i \in \mathcal{G}$, $\boldsymbol{U}_{i\oplus\left(-\frac{1}{2}\right)} \in \mathcal{G}$, and $\boldsymbol{U}_{i\oplus\frac{1}{2}} \in \mathcal{G}$. It follows that

$$\begin{aligned}
\overline{\boldsymbol{U}}_i + \Delta t \mathcal{L}_U(\boldsymbol{U}_h(t), i) &= \overline{\boldsymbol{U}}_i - \frac{\Delta t}{\Delta x}\left(\hat{\boldsymbol{F}}_{i+1/2} - \hat{\boldsymbol{F}}_{i-1/2}\right) \\
&= (1-2\hat{\omega}_1)\boldsymbol{\Pi}_i + \hat{\omega}_1 \boldsymbol{U}_{i\oplus\left(-\frac{1}{2}\right)} + \hat{\omega}_1 \boldsymbol{U}_{i\oplus\frac{1}{2}} \\
&\quad - \frac{\Delta t}{2\Delta x}\Big\{ \left[\boldsymbol{F}_1\left(\boldsymbol{U}_{i\oplus\frac{1}{2}}\right) + \boldsymbol{F}_1\left(\boldsymbol{U}_{(i+1)\oplus\left(-\frac{1}{2}\right)}\right) - \alpha\left(\boldsymbol{U}_{(i+1)\oplus\left(-\frac{1}{2}\right)} - \boldsymbol{U}_{i\oplus\frac{1}{2}}\right)\right] \\
&\quad - \left[\boldsymbol{F}_1\left(\boldsymbol{U}_{i\oplus\left(-\frac{1}{2}\right)}\right) + \boldsymbol{F}_1\left(\boldsymbol{U}_{(i-1)\oplus\frac{1}{2}}\right) - \alpha\left(\boldsymbol{U}_{i\oplus\left(-\frac{1}{2}\right)} - \boldsymbol{U}_{(i-1)\oplus\frac{1}{2}}\right)\right]\Big\} \\
&= (1-2\hat{\omega}_1)\boldsymbol{\Pi}_i + \left(\hat{\omega}_1 - \frac{\alpha\Delta t}{\Delta x}\right)\boldsymbol{U}_{i\oplus\left(-\frac{1}{2}\right)} + \left(\hat{\omega}_1 - \frac{\alpha\Delta t}{\Delta x}\right)\boldsymbol{U}_{i\oplus\frac{1}{2}} \\
&\quad + \frac{\alpha\Delta t}{2\Delta x}\left[\boldsymbol{F}_1^-(\boldsymbol{U}_{(i+1)\oplus\left(-\frac{1}{2}\right)}, \alpha) + \boldsymbol{F}_1^-(\boldsymbol{U}_{i\oplus\frac{1}{2}}, \alpha)\right] \\
&\quad + \frac{\alpha\Delta t}{2\Delta x}\left[\boldsymbol{F}_1^+(\boldsymbol{U}_{(i-1)\oplus\frac{1}{2}}, \alpha) + \boldsymbol{F}_1^+(\boldsymbol{U}_{i\oplus\left(-\frac{1}{2}\right)}, \alpha)\right].
\end{aligned}$$

Under the CFL condition (3.13), $\overline{\boldsymbol{U}}_i + \Delta t \mathcal{L}_U(\boldsymbol{U}_h(t), i)$ has been reformulated into a convex combination form. Thanks to Lemma 3.1 and the convexity of $\mathcal{G}$, we have $\overline{\boldsymbol{U}}_i + \Delta t \mathcal{L}_U(\boldsymbol{U}_h(t), i) \in \mathcal{G}$ for all i. The proof is completed. □

Theorem 3.1 implies that the proposed HWENO scheme is PCP if the forward Euler method is used for time discretization. Since the SSP Runge–Kutta method (3.10) is formally a convex combination of forward Euler, the PCP property remains valid for the fully discrete HWENO scheme (3.10), due to the convexity of $\mathcal{G}$.

4. Two-dimensional PCP HWENO scheme

In this section, we present the PCP finite volume HWENO scheme for the 2D special RHD equations

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}_1(\boldsymbol{U})}{\partial x} + \frac{\partial \boldsymbol{F}_2(\boldsymbol{U})}{\partial y} = \boldsymbol{0}, \tag{4.1}$$

where

$$\boldsymbol{U} = \left(D, m_1, m_2, E\right)^\top, \quad \boldsymbol{F}_1 = \left(Dv_1, m_1 v_1 + p, m_2 v_1, m_1\right)^\top, \quad \boldsymbol{F}_2 = \left(Dv_2, m_1 v_2 + p, m_2 v_2 + p, m_2\right)^\top.$$

Divide the computational domain into $N_x \times N_y$ uniform cells $I_{i,j} = \left[x_{i-1/2}, x_{i+1/2}\right] \times \left[y_{j-1/2}, y_{j+1/2}\right]$, $1 \le i \le N_x$, $1 \le j \le N_y$, with the cell center $(x_i, y_j) = \left(\frac{x_{i-1/2}+x_{i+1/2}}{2}, \frac{y_{j-1/2}+y_{j+1/2}}{2}\right)$. Let $\Delta x = x_{i+1/2} - x_{i-1/2}$ and $\Delta y = y_{j+1/2} - y_{j-1/2}$ denote the spatial step-sizes in the x- and y-directions, respectively.

4.1. 2D PCP finite volume HWENO scheme

In this subsection, we give the 2D finite volume PCP HWENO scheme. For the notational convenience, we denote $x_{i+a} = x_i + a\Delta x$ and $y_{j+b} = y_j + b\Delta x$, where $a \in [-\frac{1}{2}, \frac{1}{2}]$ and $b \in [-\frac{1}{2}, \frac{1}{2}]$ are real numbers.

The semi-discrete finite volume HWENO scheme for the RHD equations (4.1) is given by

$$\frac{\mathrm{d}\overline{\boldsymbol{U}}_{i,j}(t)}{\mathrm{d}t} = -\frac{1}{\Delta x}\sum_{\ell=1}^{3}\omega_\ell\left(\hat{\boldsymbol{F}}^1_{i+\frac{1}{2},j+b_\ell} - \hat{\boldsymbol{F}}^1_{i-\frac{1}{2},j+b_\ell}\right) - \frac{1}{\Delta y}\sum_{\ell=1}^{3}\omega_\ell\left(\hat{\boldsymbol{F}}^2_{i+b_\ell,j+\frac{1}{2}} - \hat{\boldsymbol{F}}^2_{i+b_\ell,j-\frac{1}{2}}\right) := \mathcal{L}_U(\boldsymbol{U}_h(t), i, j), \tag{4.2}$$

$$\begin{aligned}
\frac{\mathrm{d}\overline{\boldsymbol{V}}_{i,j}(t)}{\mathrm{d}t} &= -\frac{1}{2\Delta x}\sum_{\ell=1}^{3}\omega_\ell\left(\hat{\boldsymbol{F}}^1_{i+\frac{1}{2},j+b_\ell} + \hat{\boldsymbol{F}}^1_{i-\frac{1}{2},j+b_\ell}\right) - \frac{1}{\Delta y}\sum_{\ell=1}^{3}\omega_\ell b_\ell\left(\hat{\boldsymbol{F}}^2_{i+b_\ell,j+\frac{1}{2}} - \hat{\boldsymbol{F}}^2_{i+b_\ell,j-\frac{1}{2}}\right) \\
&\quad + \frac{1}{\Delta x}\sum_{k=1}^{3}\sum_{\ell=1}^{3}\omega_k\omega_\ell \boldsymbol{F}_1\left(\boldsymbol{U}_{i\oplus b_\ell, j\oplus b_k}\right) := \mathcal{L}_V(\boldsymbol{U}_h(t), i, j),
\end{aligned} \tag{4.3}$$

$$\frac{\mathrm{d}\overline{\boldsymbol{W}}_{i,j}(t)}{\mathrm{d}t} = -\frac{1}{\Delta x}\sum_{\ell=1}^{3}\omega_\ell b_\ell\left(\hat{\boldsymbol{F}}^1_{i+\frac{1}{2},j+b_\ell} - \hat{\boldsymbol{F}}^1_{i-\frac{1}{2},j+b_\ell}\right) - \frac{1}{2\Delta y}\sum_{\ell=1}^{3}\omega_\ell\left(\hat{\boldsymbol{F}}^2_{i+b_\ell,j+\frac{1}{2}} + \hat{\boldsymbol{F}}^2_{i+b_\ell,j-\frac{1}{2}}\right)$$

$$+\frac{1}{\Delta y}\sum_{k=1}^{3}\sum_{\ell=1}^{3}\omega_k\omega_\ell \boldsymbol{F}_2\left(\boldsymbol{U}_{i\oplus b_\ell,j\oplus b_k}\right):=\mathcal{L}_W(\boldsymbol{U}_h(t),i,j), \tag{4.4}$$

where

$$\overline{\boldsymbol{U}}_{i,j}(t)\approx\frac{1}{\Delta x\Delta y}\int_{I_{i,j}}\boldsymbol{U}(x,y,t)\,\mathrm{d}x\mathrm{d}y$$

denotes the approximation to the zeroth order moment in $I_{i,j}$,

$$\overline{\boldsymbol{V}}_{i,j}(t)\approx\frac{1}{\Delta x\Delta y}\int_{I_{i,j}}\boldsymbol{U}(x,y,t)\,\frac{x-x_i}{\Delta x}\mathrm{d}x\mathrm{d}y,$$
$$\overline{\boldsymbol{W}}_{i,j}(t)\approx\frac{1}{\Delta x\Delta y}\int_{I_{i,j}}\boldsymbol{U}(x,y,t)\,\frac{y-y_j}{\Delta x}\mathrm{d}x\mathrm{d}y,$$

denote the approximations to the first order moments in the x- and y-directions in $I_{i,j}$, respectively,

$$\boldsymbol{U}_h(t):=\{\overline{\boldsymbol{U}}_{i,j}(t)\}_{1\le i\le N_x,1\le j\le N_y}\cup\{\overline{\boldsymbol{V}}_{i,j}(t)\}_{1\le i\le N_x,1\le j\le N_y}\cup\{\overline{\boldsymbol{W}}_{i,j}(t)\}_{1\le i\le N_x,1\le j\le N_y}$$

denotes the set that contains all cells' zeroth and first order moments, and $\boldsymbol{U}_{i\oplus b_\ell,j\oplus b_k}$ denotes the approximation to $\boldsymbol{U}\left(x_{i+b_\ell},y_{j+b_k},t\right)$ within the cell $I_{i,j}$. Here we follow [62] and use the three-point Gauss quadrature with the quadrature weights and nodes given by

$$\omega_1=\omega_3=\frac{5}{18},\qquad \omega_2=\frac{4}{9},$$
$$\{b_\ell\}_{\ell=1}^3=\left\{-\frac{\sqrt{15}}{10},\ 0,\ \frac{\sqrt{15}}{10}\right\}.$$

In (4.2)–(4.4), $\hat{\boldsymbol{F}}^1_{i+\frac12,j+b_\ell}$ and $\hat{\boldsymbol{F}}^2_{i+b_\ell,j+\frac12}$ denote the numerical flux at the cell interface points $(x_{i+\frac12},y_{j+b_\ell})$ and $(x_{i+b_\ell},y_{j+\frac12})$, respectively. In this paper, we employ the Lax–Friedrichs numerical fluxes

$$\begin{cases}\hat{\boldsymbol{F}}^1_{i+\frac12,j+b_\ell}=\frac12\left(\boldsymbol{F}_1\left(\boldsymbol{U}_{i\oplus\frac12,j\oplus b_\ell}\right)+\boldsymbol{F}_1\left(\boldsymbol{U}_{(i+1)\oplus\left(-\frac12\right),j\oplus b_\ell}\right)-\alpha_1\left(\boldsymbol{U}_{(i+1)\oplus\left(-\frac12\right),j\oplus b_\ell}-\boldsymbol{U}_{i\oplus\frac12,j\oplus b_\ell}\right)\right),\\ \hat{\boldsymbol{F}}^2_{i+b_\ell,j+\frac12}=\frac12\left(\boldsymbol{F}_2\left(\boldsymbol{U}_{i\oplus b_\ell,j\oplus\frac12}\right)+\boldsymbol{F}_2\left(\boldsymbol{U}_{i\oplus b_\ell,(j+1)\oplus\left(-\frac12\right)}\right)-\alpha_2\left(\boldsymbol{U}_{i\oplus b_\ell,(j+1)\oplus\left(-\frac12\right)}-\boldsymbol{U}_{i\oplus b_\ell,j\oplus\frac12}\right)\right),\end{cases} \tag{4.5}$$

which will be useful for achieving the PCP property of our HWENO scheme. Here

$$\begin{cases}\alpha_1=\max\limits_{i,j,\ell}\left\{\max\left\{\varrho_1\left(\boldsymbol{U}_{i\oplus\frac12,j\oplus b_\ell}\right),\varrho_1\left(\boldsymbol{U}_{(i+1)\oplus\left(-\frac12\right),j\oplus b_\ell}\right)\right\}\right\},\\ \alpha_2=\max\limits_{i,j,\ell}\left\{\max\left\{\varrho_2\left(\boldsymbol{U}_{i\oplus b_\ell,j\oplus\frac12}\right),\varrho_2\left(\boldsymbol{U}_{i\oplus b_\ell,(j+1)\oplus\left(-\frac12\right)}\right)\right\}\right\},\end{cases} \tag{4.6}$$

with $\varrho_1(\boldsymbol{U})$ and $\varrho_2(\boldsymbol{U})$ denoting the spectral radius of the Jacobian matrices $\frac{\partial\boldsymbol{F}_1(\boldsymbol{U})}{\partial\boldsymbol{U}}$ and $\frac{\partial\boldsymbol{F}_2(\boldsymbol{U})}{\partial\boldsymbol{U}}$, respectively.

To compute $\mathcal{L}_U(\boldsymbol{U}_h(t),i)$, $\mathcal{L}_V(\boldsymbol{U}_h(t),i)$ and $\mathcal{L}_W(\boldsymbol{U}_h(t),i)$ in (4.2)–(4.4), one needs to reconstruct point values $\{\boldsymbol{U}_{i\oplus b_\ell,j\oplus b_k}\}_{\ell,k=1}^3$, $\{\boldsymbol{U}_{i\oplus b_\ell,j\oplus\left(\pm\frac12\right)}\}_{\ell=1}^3$ and $\{\boldsymbol{U}_{i\oplus\left(\pm\frac12\right),j\oplus b_\ell}\}_{\ell=1}^3$ by using $\{\overline{\boldsymbol{U}}_{i,j}\}$ and $\{\overline{\boldsymbol{V}}_{i,j}\}$.

For simplicity, denote $\overline{\boldsymbol{U}}_{i,j,r}$ as the zeroth-order moment in $I^r_{i,j}$ (see Fig. 3), and $\overline{\boldsymbol{V}}_{i,j,r}$, $\overline{\boldsymbol{W}}_{i,j,r}$ as the first-order moments in x-direction and y-direction in $I^r_{i,j}$. Define

$$\boldsymbol{U}^S_{i,j}=\left[\overline{\boldsymbol{U}}_{i,j,1}\ \cdots\ \overline{\boldsymbol{U}}_{i,j,9}\right],$$
$$\boldsymbol{V}^S_{i,j}=\left[\overline{\boldsymbol{V}}_{i,j,2}\ \overline{\boldsymbol{V}}_{i,j,4}\ \overline{\boldsymbol{V}}_{i,j,5}\ \overline{\boldsymbol{V}}_{i,j,6}\ \overline{\boldsymbol{V}}_{i,j,8}\right],$$
$$\boldsymbol{W}^S_{i,j}=\left[\overline{\boldsymbol{W}}_{i,j,2}\ \overline{\boldsymbol{W}}_{i,j,4}\ \overline{\boldsymbol{W}}_{i,j,5}\ \overline{\boldsymbol{W}}_{i,j,6}\ \overline{\boldsymbol{W}}_{i,j,8}\right].$$

The 2D HWENO reconstruction procedure is also based on two operators $\boldsymbol{M}_L$ and $\boldsymbol{M}_H$, which are introduced in Appendices A and B for better readability. The detailed PCP HWENO reconstruction procedure of our 2D HWENO scheme is summarized as follows:

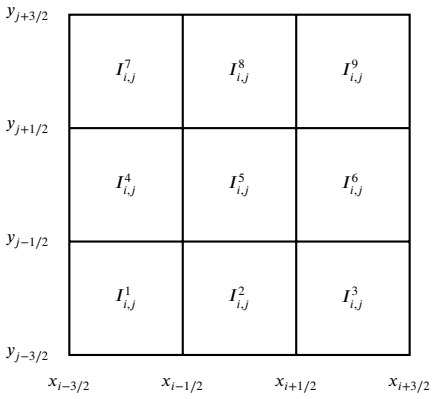


Fig. 3. Renumbering of cell $I_{i,j}$ and its adjacent cells.

Step 1. Use the KXRCF indicator [20] dimension-by-dimension to identify the troubled cells. A cell is regarded as a troubled cell as long as it is identified as troubled cell in either x-direction or y-direction. Then modify the first-order moments in the troubled cells by using the HWENO limiter given in [62].

Step 2. Reconstruct the point values $\{\boldsymbol{U}_{i\oplus b_\ell,j\oplus b_k}\}^3_{\ell,k=1}$, $\{\boldsymbol{U}_{i\oplus b_\ell,j\oplus\left(\pm\frac{1}{2}\right)}\}^3_{\ell=1}$ and $\{\boldsymbol{U}_{i\oplus\left(\pm\frac{1}{2}\right),j\oplus b_\ell}\}^3_{\ell=1}$.

- Employ the linear reconstruction for the inner Gaussian points $\{\boldsymbol{U}_{i\oplus b_\ell,j\oplus b_k}\}^3_{\ell,k=1}$:

$$\boldsymbol{U}^*_{i\oplus b_\ell,j\oplus b_k}=\boldsymbol{M}_L\left(\boldsymbol{U}^S_{i,j},\boldsymbol{V}^S_{i,j},\boldsymbol{W}^S_{i,j},b_\ell,b_k\right),\quad \forall \ell,k\in\{1,2,3\}.$$

- Reconstruction for the cell interface points $\{\boldsymbol{U}_{i\oplus b_\ell,j\oplus\left(\pm\frac{1}{2}\right)}\}^3_{\ell=1}$ and $\{\boldsymbol{U}_{i\oplus\left(\pm\frac{1}{2}\right),j\oplus b_\ell}\}^3_{\ell=1}$.

  (i) If $I_{i,j}$ is not a troubled cell, perform linear reconstruction for the cell interface points:

$$\boldsymbol{U}^*_{i\oplus\left(\pm\frac{1}{2}\right),j\oplus b_\ell}=\boldsymbol{M}_L\left(\boldsymbol{U}^S_{i,j},\boldsymbol{V}^S_{i,j},\boldsymbol{W}^S_{i,j},\pm\frac{1}{2},b_\ell\right),\quad \forall \ell\in\{1,2,3\},$$

$$\boldsymbol{U}^*_{i\oplus b_\ell,j\oplus\left(\pm\frac{1}{2}\right)}=\boldsymbol{M}_L\left(\boldsymbol{U}^S_{i,j},\boldsymbol{V}^S_{i,j},\boldsymbol{W}^S_{i,j},b_\ell,\pm\frac{1}{2}\right),\quad \forall \ell\in\{1,2,3\}.$$

  (ii) If $I_{i,j}$ is a troubled cell, perform HWENO reconstruction based on characteristic decomposition for the cell interface points:

$$\boldsymbol{U}^*_{i\oplus\left(\pm\frac{1}{2}\right),j\oplus b_\ell}=\boldsymbol{R}^1_{i\pm\frac{1}{2},j}\boldsymbol{M}_H\left(\left(\boldsymbol{R}^1_{i\pm\frac{1}{2},j}\right)^{-1}\boldsymbol{U}^S_{i,j},\left(\boldsymbol{R}^1_{i\pm\frac{1}{2},j}\right)^{-1}\boldsymbol{V}^S_{i,j},\left(\boldsymbol{R}^1_{i\pm\frac{1}{2},j}\right)^{-1}\boldsymbol{W}^S_{i,j},\pm\frac{1}{2},b_\ell\right),$$

$$\boldsymbol{U}^*_{i\oplus b_\ell,j\oplus\left(\pm\frac{1}{2}\right)}=\boldsymbol{R}^2_{i,j\pm\frac{1}{2}}\boldsymbol{M}_H\left(\left(\boldsymbol{R}^2_{i,j\pm\frac{1}{2}}\right)^{-1}\boldsymbol{U}^S_{i,j},\left(\boldsymbol{R}^2_{i,j\pm\frac{1}{2}}\right)^{-1}\boldsymbol{V}^S_{i,j},\left(\boldsymbol{R}^2_{i,j\pm\frac{1}{2}}\right)^{-1}\boldsymbol{W}^S_{i,j},b_\ell,\pm\frac{1}{2}\right),$$

for all $\ell\in\{1,2,3\}$. $\left(\boldsymbol{R}^1_{i+\frac{1}{2},j}\right)^{-1}$ and $\boldsymbol{R}^1_{i+\frac{1}{2},j}$ are taken as left and right eigenvector matrices of $\boldsymbol{A}_1\left(\overline{\boldsymbol{U}}_{i,j},\overline{\boldsymbol{U}}_{i+1,j}\right)$, which is the Roe matrix [9] associated with $\frac{\partial \boldsymbol{F}_1}{\partial \boldsymbol{U}}$, $\left(\boldsymbol{R}^2_{i,j+\frac{1}{2}}\right)^{-1}$ and $\boldsymbol{R}^2_{i,j+\frac{1}{2}}$ are taken as left and right eigenvector matrices of $\boldsymbol{A}_2\left(\overline{\boldsymbol{U}}_{i,j},\overline{\boldsymbol{U}}_{i,j+1}\right)$, which is the Roe matrix associated with $\frac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{U}}$. The eigenvector matrices and fluxes are computed by using the primitive variables, which are recovered by using the proposed NR methods.

Step 3. Perform the PCP limiter on the reconstructed point values $\{\boldsymbol{U}^*_{i\oplus b_\ell,j\oplus b_k}\}^3_{\ell,k=1}$, $\{\boldsymbol{U}^*_{i\oplus\left(\pm\frac{1}{2}\right),j\oplus b_\ell}\}^3_{\ell=1}$, and $\{\boldsymbol{U}^*_{i\oplus b_\ell,j\oplus\left(\pm\frac{1}{2}\right)}\}^3_{\ell=1}$ as follows. Define $\Theta:=\{(b_\ell,b_k)\}^3_{\ell,k=1}\cup\{(\pm\frac{1}{2},b_\ell)\}^3_{\ell=1}\cup\{(b_\ell,\pm\frac{1}{2})\}^3_{\ell=1}$ and

$$\boldsymbol{U}^*_{i\oplus a,j\oplus b}=[D^*_{i\oplus a,j\oplus b}\ (m_1)^*_{i\oplus a,j\oplus b}\ (m_2)^*_{i\oplus a,j\oplus b}\ E^*_{i\oplus a,j\oplus b}]^\top,\quad \forall(a,b)\in\Theta.$$

Denote the first component of $\overline{\boldsymbol{U}}_{i,j}$ as $\overline{D}_{i,j}$. Let

$$\mu_1=\frac{\lambda_1\alpha_1}{\lambda_1\alpha_1+\lambda_2\alpha_2},\qquad \mu_2=\frac{\lambda_2\alpha_2}{\lambda_1\alpha_1+\lambda_2\alpha_2)}$$

with $\lambda_1=\Delta t/\Delta x$, $\lambda_2=\Delta t/\Delta y$.

- Modify the mass density to enforce its positivity via

$$\tilde{D}_{i\oplus a,j\oplus b}=\theta_D(D^*_{i\oplus a,j\oplus b}-\overline{D}_{i,j})+\overline{D}_{i,j}\quad\text{with}\quad\theta_D=\min\left\{\left|\frac{\overline{D}_{i,j}-\epsilon_D}{\overline{D}_{i,j}-D_{\min}}\right|,1\right\},$$

$$D_{\min}=\min\left\{\min_{\ell}\left\{D^*_{i\oplus\left(\pm\frac{1}{2}\right),j\oplus b_\ell}\right\},\min_{\ell}\left\{D^*_{i\oplus b_\ell,j\oplus\left(\pm\frac{1}{2}\right)}\right\},\min_{\ell,k}\left\{D^*_{i\oplus b_\ell,j\oplus b_k}\right\},\widehat{D}_{i,j}\right\},$$

$$\widehat{D}_{i,j}:=\frac{1}{1-2\hat{\omega}_1}\left(\overline{D}_{i,j}-\sum_{\ell=1}^{3}\omega_\ell\hat{\omega}_1\left[\mu_1\left(D^*_{i\oplus\frac{1}{2},j\oplus b_\ell}+D^*_{i\oplus\left(-\frac{1}{2}\right),j\oplus b_\ell}\right)+\mu_2\left(D^*_{i\oplus b_\ell,j\oplus\frac{1}{2}}+D^*_{i\oplus b_\ell,j\oplus\left(-\frac{1}{2}\right)}\right)\right]\right),$$

where $\epsilon_D=\min_{i,j}\left\{10^{-13},\overline{D}_{i,j}\right\}$ is a small positive number introduced to avoid the effect of round-off errors.

- Define $\tilde{\boldsymbol{U}}_{i\oplus a,j\oplus b}=[\tilde{D}_{i\oplus a,j\oplus b}\ (m_1)^*_{i\oplus a,j\oplus b}\ (m_2)^*_{i\oplus a,j\oplus b}\ E^*_{i\oplus a,j\oplus b}]^\top$ for all $(a,b)\in\Theta$. Enforce the positivity of $g(\boldsymbol{U})$ by

$$\boldsymbol{U}_{i\oplus a,j\oplus b}=\theta_g(\tilde{\boldsymbol{U}}_{i+a,j+b}-\overline{\boldsymbol{U}}_{i,j})+\overline{\boldsymbol{U}}_{i,j}\quad\text{with}\quad\theta_g=\min\left\{\left|\frac{g(\overline{\boldsymbol{U}}_{i,j})-\epsilon_g}{g(\overline{\boldsymbol{U}}_{i,j})-g_{\min}}\right|,1\right\},\tag{4.7}$$

$$g_{\min}=\min\left\{\min_{\ell}\left\{g(\tilde{\boldsymbol{U}}_{i\oplus\left(\pm\frac{1}{2}\right),j\oplus b_\ell})\right\},\min_{\ell}\left\{g(\tilde{\boldsymbol{U}}_{i\oplus b_\ell,j\oplus\left(\pm\frac{1}{2}\right)})\right\},\min_{\ell,k}\left\{g(\tilde{\boldsymbol{U}}_{i\oplus b_\ell,j\oplus b_k})\right\},g(\widetilde{\boldsymbol{\Pi}}_{i,j})\right\},$$

$$\widetilde{\boldsymbol{\Pi}}_{i,j}:=\frac{1}{1-2\hat{\omega}_1}\left(\overline{\boldsymbol{U}}_{i,j}-\sum_{\ell=1}^{3}\omega_\ell\hat{\omega}_1\left[\mu_1\left(\tilde{\boldsymbol{U}}_{i\oplus\frac{1}{2},j\oplus b_\ell}+\tilde{\boldsymbol{U}}_{i\oplus\left(-\frac{1}{2}\right),j\oplus b_\ell}\right)+\mu_2\left(\tilde{\boldsymbol{U}}_{i\oplus b_\ell,j\oplus\frac{1}{2}}+\tilde{\boldsymbol{U}}_{i\oplus b_\ell,j\oplus\left(-\frac{1}{2}\right)}\right)\right]\right),$$

where $\epsilon_g=\min_{i,j}\left\{10^{-13},g(\overline{\boldsymbol{U}}_{i,j})\right\}$.

4.2. Rigorous analysis of PCP property

In this subsection, we present a rigorous analysis of PCP property of the proposed 2D HWENO scheme.
Similar to [4, Proposition 3.1], we can prove that the PCP limited point values (4.7) satisfy the following property.

Lemma 4.1. *If $\overline{\boldsymbol{U}}_{i,j}\in\mathcal{G}$, then:*

(1) $\boldsymbol{U}_{i\oplus\left(\pm\frac{1}{2}\right),j\oplus b_\ell}\in\mathcal{G}$, $\boldsymbol{U}_{i\oplus b_\ell,j\oplus\left(\pm\frac{1}{2}\right)}\in\mathcal{G}$ *and* $\boldsymbol{U}_{i\oplus b_\ell,j\oplus b_k}\in\mathcal{G}$ *for all* $\ell,k\in\{1,2,3\}$.
(2) $\boldsymbol{\Pi}_{i,j}\in\mathcal{G}$, *where*

$$\boldsymbol{\Pi}_{i,j}:=\frac{1}{1-2\hat{\omega}_1}\left(\overline{\boldsymbol{U}}_{i,j}-\sum_{\beta=1}^{3}\omega_\beta\hat{\omega}_1\left[\mu_1\left(\boldsymbol{U}_{i\oplus\frac{1}{2},j\oplus b_\beta}+\boldsymbol{U}_{i\oplus\left(-\frac{1}{2}\right),j\oplus b_\beta}\right)+\mu_2\left(\boldsymbol{U}_{i\oplus b_\beta,j\oplus\frac{1}{2}}+\boldsymbol{U}_{i\oplus b_\beta,j\oplus\left(-\frac{1}{2}\right)}\right)\right]\right).\tag{4.8}$$

Base on the Lemmas 3.1 and 4.1, we are now ready to show the PCP property of the proposed 2D HWENO scheme.

Theorem 4.1. *Consider the proposed 2D HWENO scheme (4.2)–(4.4) with the Lax–Friedrichs flux (4.5). If $\overline{\boldsymbol{U}}_{i,j}\in\mathcal{G}$ for all i, j, then the updated cell averages*

$$\overline{\boldsymbol{U}}_{i,j}+\Delta t\mathcal{L}_U(\boldsymbol{U}_h(t),i,j)\in\mathcal{G},\qquad\forall i,j\tag{4.9}$$

under the CFL condition

$$\Delta t\le\frac{\hat{\omega}_1}{\alpha_1/\Delta x+\alpha_2/\Delta y}.\tag{4.10}$$

Proof. Based on (4.8), we have the following convex decomposition for the cell average:

$$\overline{\boldsymbol{U}}_{i,j}=(1-2\hat{\omega}_1)\boldsymbol{\Pi}_{i,j}+\sum_{\beta=1}^{3}\omega_\beta\hat{\omega}_1\left[\mu_1\left(\boldsymbol{U}_{i\oplus\frac{1}{2},j\oplus b_\beta}+\boldsymbol{U}_{i\oplus\left(-\frac{1}{2}\right),j\oplus b_\beta}\right)+\mu_2\left(\boldsymbol{U}_{i\oplus b_\beta,j\oplus\frac{1}{2}}+\boldsymbol{U}_{i\oplus b_\beta,j\oplus\left(-\frac{1}{2}\right)}\right)\right]$$

with $\boldsymbol{\Pi}_{i,j}\in\mathcal{G}$, $\boldsymbol{U}_{i\oplus\left(\pm\frac{1}{2}\right),j\oplus b_\ell}\in\mathcal{G}$, $\boldsymbol{U}_{i\oplus b_\ell,j\oplus\left(\pm\frac{1}{2}\right)}\in\mathcal{G}$, and $\boldsymbol{U}_{i\oplus b_\ell,j\oplus b_k}\in\mathcal{G}$ for all $\ell,k\in\{1,2,3\}$. It follows that

$$
\begin{aligned}
&\overline{\boldsymbol{U}}_{i,j}+\Delta t\mathcal{L}_U(\boldsymbol{U}_h(t),i,j)\\
=&(1-2\hat{\omega}_1)\boldsymbol{\Pi}_{i,j}+\sum_{\ell=1}^{3}\omega_\ell\hat{\omega}_1\left[\mu_1\left(\boldsymbol{U}_{i\oplus\frac{1}{2},j\oplus b_\ell}+\boldsymbol{U}_{i\oplus\left(-\frac{1}{2}\right),j\oplus b_\ell}\right)+\mu_2\left(\boldsymbol{U}_{i\oplus b_\ell,j\oplus\frac{1}{2}}+\boldsymbol{U}_{i\oplus b_\ell,j\oplus\left(-\frac{1}{2}\right)}\right)\right]\\
&-\frac{\Delta t}{\Delta x}\sum_{\ell=1}^{3}\omega_\ell\left(\hat{\boldsymbol{F}}^1_{i+\frac{1}{2},j+b_\ell}-\hat{\boldsymbol{F}}^1_{i-\frac{1}{2},j+b_\ell}\right)-\frac{\Delta t}{\Delta y}\sum_{\ell=1}^{3}\omega_\ell\left(\hat{\boldsymbol{F}}^2_{i+b_\ell,j+\frac{1}{2}}-\hat{\boldsymbol{F}}^2_{i+b_\ell,j-\frac{1}{2}}\right)\\
=&(1-2\hat{\omega}_1)\boldsymbol{\Pi}_{i,j}+\mu_1\sum_{\ell=1}^{3}\omega_\ell\hat{\omega}_1\left[\left(1-\frac{\lambda_1\alpha_1}{\mu_1\hat{\omega}_1}\right)(\boldsymbol{U}_{i\oplus\left(-\frac{1}{2}\right),j\oplus b_\ell}+\boldsymbol{U}_{i\oplus\frac{1}{2},j\oplus b_\ell})+\right.\\
&\left.\frac{\lambda_1\alpha_1}{2\mu_1\hat{\omega}_1}\left(\boldsymbol{F}_1^+(\boldsymbol{U}_{(i-1)\oplus\frac{1}{2},j\oplus b_\ell},\alpha_1)+\boldsymbol{F}_1^+(\boldsymbol{U}_{i\oplus\left(-\frac{1}{2}\right),j\oplus b_\ell},\alpha_1)+\boldsymbol{F}_1^-(\boldsymbol{U}_{i\oplus\frac{1}{2},j\oplus b_\ell},\alpha_1)+\boldsymbol{F}_1^-(\boldsymbol{U}_{(i+1)\oplus\left(-\frac{1}{2}\right),j\oplus b_\ell},\alpha_1)\right)\right]\\
&+\mu_2\sum_{\ell=1}^{3}\omega_\ell\hat{\omega}_1\left[\left(1-\frac{\lambda_2\alpha_2}{\mu_2\hat{\omega}_1}\right)(\boldsymbol{U}_{i\oplus b_\ell,j\oplus\left(-\frac{1}{2}\right)}+\boldsymbol{U}_{i\oplus b_\ell,j\oplus\frac{1}{2}})+\right.\\
&\left.\frac{\lambda_2\alpha_2}{2\mu_2\hat{\omega}_1}\left(\boldsymbol{F}_2^+(\boldsymbol{U}_{i\oplus b_\ell,(j-1)\oplus\frac{1}{2}},\alpha_2)+\boldsymbol{F}_2^+(\boldsymbol{U}_{i\oplus b_\ell,j\oplus\left(-\frac{1}{2}\right)},\alpha_2)+\boldsymbol{F}_2^-(\boldsymbol{U}_{i\oplus b_\ell,j\oplus\frac{1}{2}},\alpha_2)+\boldsymbol{F}_2^-(\boldsymbol{U}_{i\oplus b_\ell,(j+1)\oplus\left(-\frac{1}{2}\right)},\alpha_2)\right)\right].
\end{aligned}
$$

Under the CFL condition (4.10), $\overline{\boldsymbol{U}}_{i,j}+\Delta t\mathcal{L}_U(\boldsymbol{U}_h(t),i,j)$ has been reformulated into a convex combination form. Thanks to Lemma 3.1 and the convexity of $\mathcal{G}$, we obtain (4.9). The proof is completed. □

Theorem 4.1 implies that the proposed 2D HWENO scheme is PCP if the forward Euler method is used for time discretization. Since the SSP Runge–Kutta method (3.10) is formally a convex combination of forward Euler, the PCP property remains valid for the fully discrete HWENO scheme, due to the convexity of $\mathcal{G}$.

4.3. Application to axisymmetric RHD equations in cylindrical coordinates

In this subsection, we present the HWENO scheme for the axisymmetric RHD equations in cylindrical coordinates (r,z), which can be written as

$$
\frac{\partial\boldsymbol{U}}{\partial t}+\frac{\partial\boldsymbol{F}_1(\boldsymbol{U})}{\partial r}+\frac{\partial\boldsymbol{F}_2(\boldsymbol{U})}{\partial z}=\boldsymbol{S}(\boldsymbol{U},r),
$$

where the definition of fluxes $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ are the same as (1.3), and the source term

$$
\boldsymbol{S}(\boldsymbol{U},r)=-\frac{1}{r}(Dv_1,m_1v_1,m_2v_1,m_1)^\top.
$$

The semi-discrete HWENO scheme for axisymmetric RHD equations reads

$$
\begin{cases}
\dfrac{\mathrm{d}\overline{\boldsymbol{U}}_{i,j}}{\mathrm{d}t}=\mathcal{L}_U(\boldsymbol{U}_h(t),i,j)+\overline{\boldsymbol{S}}_{i,j},\\
\dfrac{\mathrm{d}\overline{\boldsymbol{V}}_{i,j}}{\mathrm{d}t}=\mathcal{L}_V(\boldsymbol{U}_h(t),i,j)+\overline{\boldsymbol{S}}^1_{i,j},\\
\dfrac{\mathrm{d}\overline{\boldsymbol{W}}_{i,j}}{\mathrm{d}t}=\mathcal{L}_W(\boldsymbol{U}_h(t),i,j)+\overline{\boldsymbol{S}}^2_{i,j},
\end{cases}
\tag{4.11}
$$

where

$$
\begin{aligned}
\overline{\boldsymbol{S}}_{i,j}&:=\sum_{\ell=1}^{3}\sum_{k=1}^{3}\omega_\ell\omega_k\boldsymbol{S}\left(\boldsymbol{U}_{i\oplus b_\ell,j\oplus b_k}\right),\\
\overline{\boldsymbol{S}}^1_{i,j}&:=\sum_{\ell=1}^{3}\sum_{k=1}^{3}\omega_\ell\omega_k b_\ell\boldsymbol{S}\left(\boldsymbol{U}_{i\oplus b_\ell,j\oplus b_k}\right),\\
\overline{\boldsymbol{S}}^2_{i,j}&:=\sum_{\ell=1}^{3}\sum_{k=1}^{3}\omega_\ell\omega_k b_k\boldsymbol{S}\left(\boldsymbol{U}_{i\oplus b_\ell,j\oplus b_k}\right)
\end{aligned}
$$

are the numerical approximations to $\int_{I_{i,j}}\boldsymbol{S}(\boldsymbol{U},r)\mathrm{d}r\mathrm{d}z$, $\int_{I_{i,j}}\boldsymbol{S}(\boldsymbol{U},r)\frac{r-r_i}{\Delta r}\mathrm{d}r\mathrm{d}z$, and $\int_{I_{i,j}}\boldsymbol{S}(\boldsymbol{U},r)\frac{z-z_j}{\Delta z}\mathrm{d}r\mathrm{d}z$, respectively. The definitions of spatial operators $\mathcal{L}_U$, $\mathcal{L}_V$ and $\mathcal{L}_W$ are the same as (4.2)–(4.4) (with the variables x, y replaced by r, z).

To analysis the PCP property of the scheme (4.11), we recall the following lemma proposed in [50, Section 3.2]:

Lemma 4.2. *If $\boldsymbol{U} \in \mathcal{G}$, then $\boldsymbol{U} + \Delta t \boldsymbol{S}(\boldsymbol{U}, r) \in \mathcal{G}$ under the time step restriction*

$$v_1 \Delta t \le \frac{r g(\boldsymbol{U})}{p + g(\boldsymbol{U})}.$$

Assume β is a positive number, then we have

$$\begin{aligned}
&\overline{\boldsymbol{U}}_{i,j} + \Delta t(\mathcal{L}_U(\boldsymbol{U}_h(t), i, j) + \overline{\boldsymbol{S}}_{i,j}) \\
=&(1-\beta)(\overline{\boldsymbol{U}}_{i,j} + \frac{\Delta t}{1-\beta}\mathcal{L}_U(\boldsymbol{U}(t), i, j)) + \beta(\overline{\boldsymbol{U}}_{i,j} + \frac{\Delta t}{\beta}\overline{\boldsymbol{S}}_{i,j}) \\
=&(1-\beta)\left(\overline{\boldsymbol{U}}_{i,j} + \frac{\Delta t}{1-\beta}\mathcal{L}_U(\boldsymbol{U}(t), i, j)\right) + \beta\left(\sum_{\ell=1}^{3}\sum_{k=1}^{3}\omega_\ell\omega_k\left(\boldsymbol{U}_{i\oplus b_\ell, j\oplus b_k} + \frac{\Delta t}{\beta}\boldsymbol{S}\left(\boldsymbol{U}_{i\oplus b_\ell, j\oplus b_k}\right)\right)\right).
\end{aligned}$$

By using Theorem 4.1, Lemma 4.1, Lemma 4.2, and the convexity of $\mathcal{G}$, one can deduce that the HWENO scheme (4.11) preserves

$$\overline{\boldsymbol{U}}_{i,j} + \Delta t(\mathcal{L}_U(\boldsymbol{U}_h(t), i, j) + \overline{\boldsymbol{S}}_{i,j}) \in \mathcal{G}$$

under the time step restriction

$$\frac{\Delta t}{1-\beta} \le \frac{\hat{\omega}_1}{\alpha_1/\Delta x + \alpha_2/\Delta y}, \qquad v_1(\boldsymbol{U}_{i\oplus b_\ell, j\oplus b_k})\frac{\Delta t}{\beta} \le \frac{(r_i + b_\ell \Delta r) g(\boldsymbol{U}_{i\oplus b_\ell, j\oplus b_k})}{\left(p(\boldsymbol{U}_{i\oplus b_\ell, j\oplus b_k}) + g(\boldsymbol{U}_{i\oplus b_\ell, j\oplus b_k})\right)}. \tag{4.12}$$

Taking a special β leads to the following time step restriction

$$\Delta t \le \beta A_s,$$

where

$$\begin{aligned}
A_s &= \min_{i,j}\left\{\min_{(\ell,k)\in\Lambda_{ij}}\left\{\frac{(r_i + b_\ell \Delta r) g(\boldsymbol{U}_{i\oplus b_\ell, j\oplus b_k})}{\left(p(\boldsymbol{U}_{i\oplus b_\ell, j\oplus b_k}) + g(\boldsymbol{U}_{i\oplus b_\ell, j\oplus b_k})\right) v_1(\boldsymbol{U}_{i\oplus b_\ell, j\oplus b_k})}\right\}\right\}, \\
\beta &= \frac{\hat{\omega}_1}{\hat{\omega}_1 + A_s\left(\alpha_1/\Delta x + \alpha_2/\Delta y\right)}, \qquad \Lambda_{ij} := \left\{(\ell, k) : v_1(\boldsymbol{U}_{i\oplus b_\ell, j\oplus b_k}) > 0\right\}.
\end{aligned}$$

Again, the PCP property remains valid for the fully discrete HWENO scheme if the SSP Runge–Kutta method is used for time discretization.

5. Numerical tests

This section will conduct several ultra-relativistic numerical experiments, to demonstrate the accuracy, robustness, and effectiveness of the proposed PCP finite volume HWENO schemes and NR methods. The CFL number is set as 0.6, and unless otherwise specified, the adiabatic index γ is set as $5/3$. We will compare the proposed NR methods with several other primitive variables recovery algorithms in Examples 5.1 and 5.10. In the other examples, we only present the results obtained by using our preferred hybrid NR method due to its superior efficacy.

5.1. Numerical experiments on primitive variables recovery algorithms

We conduct several tests to evaluate the accuracy and efficiency of the three proposed quadratically convergent PCP primitive variable recovering algorithms (the target accuracy ϵ_{target} is set as 10^{-14}), by comparing them with three existing algorithms from the literature, including the hybrid linearly convergent PCP algorithm introduced in [4] (hereafter termed "Hybrid-linear"), the NR algorithm proposed by Mignone, Plewa, and Bodo in [29] (which we denote as "MPB-NR"), and the velocity-proxy-based recovery algorithm from [37] (which we refer to as "Vel-Proxy"). All the tests are implemented in C++ with double precision and performed with one core on the same Windows environment with 13th Gen Intel(R) Core(TM) i9-13900HX2.20 GHz.

Example 5.1 *(Random tests).* Three sets of random tests are provided to validate the accuracy and efficiency of our NR methods presented in Section 2. Let $U_{\texttt{rand}}$ denote the uniform random variables independently generated in $[0, 1]$.

The first set of random primitive variables are generated by

$$\begin{cases}
\rho = 1000 U_{\texttt{rand}} + 10^{-10}, \\
v = 1.99999 U_{\texttt{rand}} - 1.99999/2, \\
p = 10 U_{\texttt{rand}} + 10^{-10}, \\
\gamma = 1 + U_{\texttt{rand}}.
\end{cases} \tag{5.1}$$

Table 1
The first set of random tests: CPU time, maximum relative errors, average relative errors, average iterations, algorithm failure counts, and negative pressure counts in 10^8 independent random experiments.

algorithms	total time (s)	max error	average error	average iteration	negative number	failures
Hybrid-linear	107.806	7.52E-07	2.22E-13	15.2606	0	0
NR-I	9.522	3.26E-07	1.51E-13	4.3830	0	0
NR-II	11.293	4.26E-07	1.62E-13	3.8089	0	0
Hybrid NR	9.033	3.26E-07	1.51E-13	4.3448	0	0
Analytical	32.861	7.42E-00	3.12E-07	-	0	0
Vel-Proxy	11.686	6.70E-07	1.74E-13	4.50908	0	0
MPB-NR	42.056	3.26E-07	1.65E-13	4.1707	0	0

Table 2
The second set of random tests: CPU time, maximum relative errors, average relative errors, average iterations, algorithm failure counts, and negative pressure counts in 10^8 independent random experiments.

algorithms	total time (s)	max error	average error	average iteration	negative number	failures
Hybrid-linear	123.606	3.25E-02	4.04E-10	17.8687	0	0
NR-I	19.631	2.83E-06	5.99E-13	11.0891	0	0
NR-II	11.455	4.75E-09	5.20E-15	3.5914	0	0
Hybrid NR	12.342	4.75E-09	5.52E-15	4.65143	0	0
Analytical	32.423	9.11E-00	2.71E-07	-	0	0
Vel-Proxy	11.599	5.42E-09	7.26E-15	4.50908	0	0
MPB-NR	42.662	4.75E-09	5.32E-15	4.2800	0	0

The second set of random tests involve low density and low pressure:

$$\begin{cases} \rho = 10^{-3} U_{\texttt{rand}} + 10^{-10}, \\ v = 1.99999 U_{\texttt{rand}} - 1.99999/2, \\ p = 0.1 U_{\texttt{rand}} + 10^{-10}, \\ \gamma = 1 + U_{\texttt{rand}}. \end{cases} \tag{5.2}$$

In the last set of random tests, velocity approaches the speed of light, density is small and $\gamma = 2$:

$$\begin{cases} \rho = 10^{-4}, \\ v = 1 - 10^{-8} - 10^{-6} U_{\texttt{rand}}, \\ p = 500 U_{\texttt{rand}} + 500, \\ \gamma = 2. \end{cases} \tag{5.3}$$

In our experimental setup, we generate primitive variables randomly and use them to calculate the corresponding conservative variables $\boldsymbol{U}$ through (1.4). We then apply the primitive variables recovery algorithms to recompute the primitive variables from $\boldsymbol{U}$. The selection of an initial value for the MPB-NR method [29] is important but currently lacks both theoretical and empirical guidance. Therefore, in our experiments, we introduce a 20% random perturbation to the exact pressure p to serve as our initial guess. The results of our tests, which consist of 10^8 independent random experiments, are presented in Tables 1–3. These tables provide information on the total CPU time in seconds, maximum relative errors in p, average relative errors in p, average iteration numbers, algorithm failure counts, and total numbers of negative p appearing in iterations. We observe that no negative pressure is produced in our three NR methods, confirming their robustness and PCP property. For the second test reported in Table 2, it is seen that the NR-I method may encounter the ill-posed problem discussed in Section 2.4 and takes higher CPU time compared to the NR-II, hybrid NR, and Vel-Proxy methods. The experimental results indicate that the proposed hybrid NR method overall exhibits superior performance in terms of speed, robustness, efficiency, and accuracy across all tests.

5.2. One-dimensional examples

Example 5.2 *(1D accuracy test).* This is an ultra-relativistic smooth problem that serves the purpose of testing the accuracy of the 1D PCP HWENO scheme. The initial condition is given as

$$\boldsymbol{Q}(x,0) = (1 + 0.99999\sin(x), 0.9999, 0.0001)^T, x \in [0, 2\pi).$$

Due to the low density, large velocity close to the speed of light, and low pressure, this test is challenging, and the PCP limiting produce is necessary for successful simulation. We use the PCP HWENO scheme and the finite volume WENO scheme [65] with the PCP limiter to simulate this problem on the mesh of N uniform cells with $N \in \{30, 60, 90, \ldots, 180\}$. Table 4 lists the numerical errors in the mass density D and the convergence rates in L^1, L^2 and L^∞ norms at time $t = 2\pi$. We also provide the CPU time and the

Table 3
The third set of random tests: CPU time, maximum relative errors, average relative errors, average iterations, algorithm failure counts, and negative pressure counts in 10^8 independent random experiments.

algorithms	total time (s)	max error	average error	average iteration	negative number	failures
Hybrid-linear	359.914	4.70E-01	2.64E-03	72.3186	0	0
NR-I	11.090	4.70E-01	2.64E-03	5.2111	0	0
NR-II	19.224	4.70E-01	2.64E-03	7.3604	0	0
Hybrid NR	18.778	4.70E-01	2.64E-03	7.3604	0	0
Analytical	32.472	4.70E-01	2.64E-03	-	0	0
Vel-Proxy	50.511	7.63E-01	3.69E-03	26.9903	0	0
MPB-NR	62.754	4.70E-01	2.64E-03	6.3865	4535	4535

Table 4
Example 5.2: CPU time in seconds, the percentage of PCP limited cells, the numerical errors of mass density in L^1, L^2, and L^∞ norms, and the corresponding convergence rates.

Scheme	N	L^1 error	Order	L^2 error	Order	L^∞ error	Order	Limited cells	CPU time
HWENO	30	3.56E-01	-	7.14E-01	-	2.35E-00	-	7.6%	6.4E-02s
	60	8.40E-06	15.37	2.85E-05	14.61	1.82E-04	13.66	1.6%	3.2E-01s
	90	1.11E-07	10.68	1.23E-07	13.43	1.74E-07	17.15	0%	9.7E-01s
	120	1.97E-08	6.00	2.19E-08	6.00	3.10E-08	6.00	0%	2.1E-00s
	150	5.17E-09	5.99	5.75E-09	5.99	8.21E-09	5.95	0%	4.0E-00s
	180	1.74E-09	5.97	1.94E-09	5.97	2.82E-09	5.86	0%	6.9E-00s
WENO	30	8.09E-03	-	1.29E-02	-	4.75E-02	-	34.9%	5.2E-02s
	60	2.26E-04	5.16	4.67E-04	4.79	2.08E-03	4.52	21.1%	2.6E-01s
	90	2.01E-05	5.97	3.82E-05	6.17	2.29E-04	5.44	8.7%	7.4E-01s
	120	2.79E-06	6.86	3.10E-06	8.73	4.38E-06	13.75	0%	1.6E-00s
	150	9.15E-07	5.00	1.02E-06	5.00	1.44E-06	5.00	0%	3.0E-00s
	180	3.68E-07	5.00	4.08E-07	5.00	5.78E-07	5.00	0%	5.0E-00s

percentage of PCP limited cells in Table 4. Attributed to the use of derivative data in the reconstruction, the 1D PCP HWENO scheme achieves sixth-order convergence rate, which is higher than the fifth-order convergence rate of the WENO scheme. While HWENO's CPU time is slightly higher than WENO's on the same meshes, the improved accuracy implies that, in terms of overall efficiency, HWENO outperforms WENO.

Example 5.3 *(1D Riemann problem).* The initial conditions of this problem [4] are given by

$$\boldsymbol{Q}(x,0)=\begin{cases}(10^{-2},0,1)^{\top}, & x\le 0.5,\\ (10^{-2},0,10^{-2})^{\top}, & x>0.5.\end{cases} \tag{5.4}$$

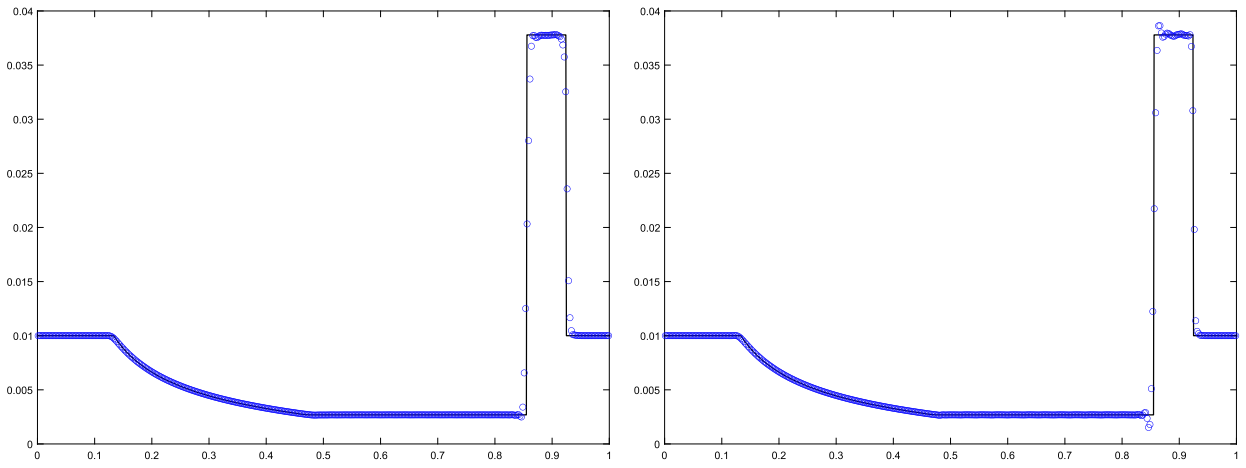
The initial discontinuity results in a rarefaction wave, a right-moving contact discontinuity, and a right-moving shock wave. This Riemann problem is used to validate the importance and effectiveness of using scale-invariant nonlinear weights for controlling spurious oscillations. Fig. 4 presents the numerical results at $t=0.45$, obtained by the PCP HWENO scheme using the scale-invariant and non-scale-invariant nonlinear weights, with 400 uniform cells in the computational domain $[0,1]$. We see that the waves are well captured using our scale-invariant nonlinear weights, while the numerical solution obtained using the non-scale-invariant nonlinear weights [61] exhibits notable overshoots and undershoots near the contact discontinuity.

Example 5.4 *(Quasi-1D Riemann problem).* This example is proposed in [56], and its initial conditions are given by

$$\boldsymbol{Q}(x,y,0)=\begin{cases}(1,0.8,0,1000)^{\top}, & x\le 0,\\ (1,0,0.999,0.01)^{\top}, & x>0.\end{cases} \tag{5.5}$$

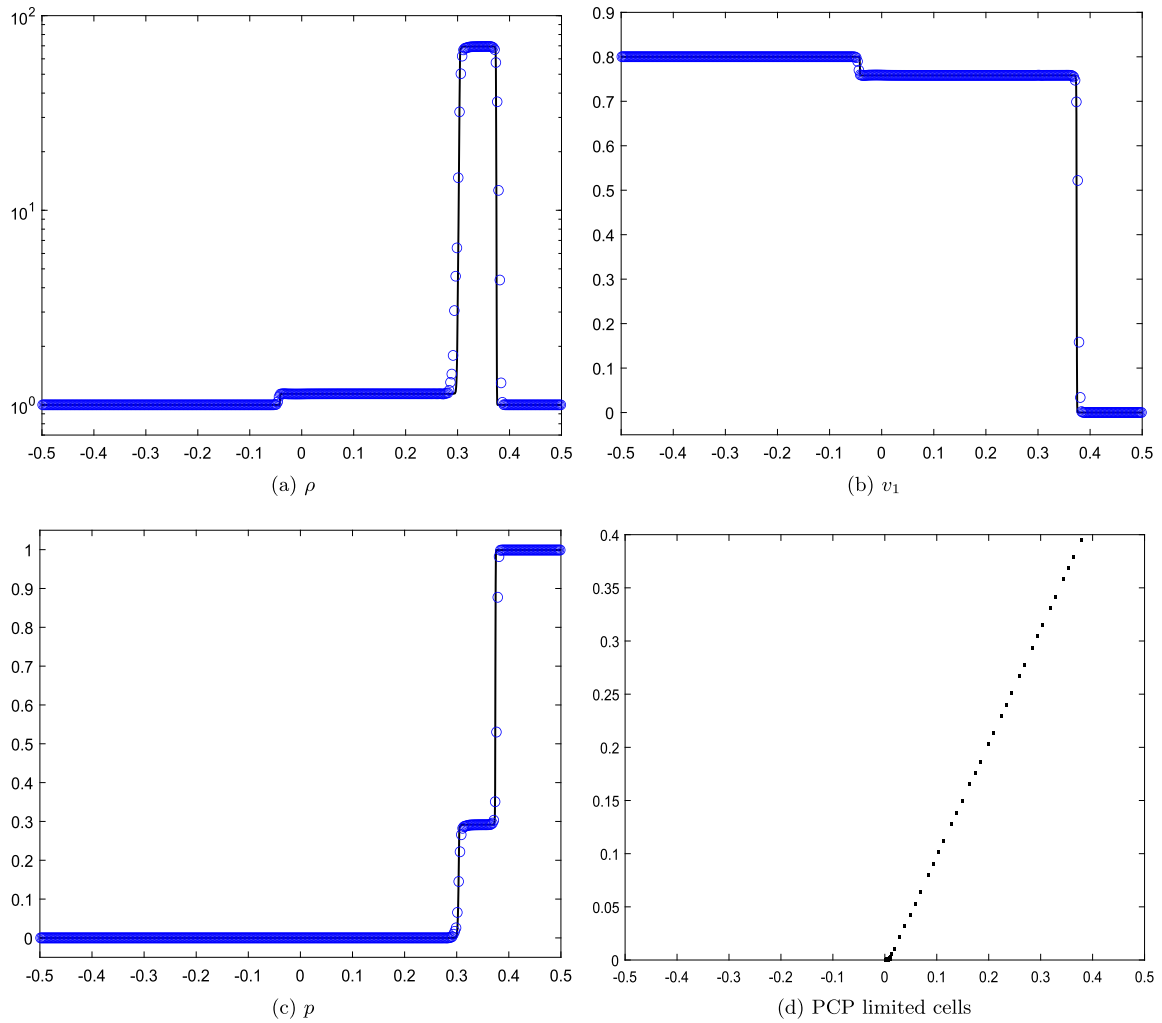
Due to the inclusion of tangential velocity, the velocity components are coupled through the Lorentz factor, leading to effects that are not presented in the non-relativistic hydrodynamics. For our simulation, we employ the 2D PCP HWENO scheme on a mesh of 400×5 uniform cells in the computational domain $[-0.5,0.5]\times[-\frac{1}{160},\frac{1}{160}]$, with $\Delta x=\Delta y=\frac{1}{400}$. Fig. 5 presents the numerical solution along the line $y=0$ at $t=0.4$. We observe that our PCP HWENO scheme remains robust and accurately captures the wave structures against the exact solution. We also notice that for this challenging test, the PCP limiter is necessary for successful simulation. If the PCP limiter is turned off, the simulation will fail immediately during the first time step due to nonphysical solutions. The PCP limited cells along $y=0$ from $t=0$ to $t=0.4$ are also displayed in Fig. 5.

Example 5.5 *(Shock heating problem).* This example simulates the shock heating problem, has become a standard test for evaluating the ability of numerical schemes to handle strong shocks without generating excessive postshock oscillations. The initial data in the computational domain $[0,1]$ is given as



(a) Use our scale-invariant nonlinear weights (b) Use the nonlinear weights proposed in [61]

Fig. 4. Example 5.3. The numerical solution (symbols "∘") and exact solution (solid lines) of density ρ obtained using our scale-invariant nonlinear weights and the nonlinear weights proposed in [61].



(a) ρ (b) v_1 (c) p (d) PCP limited cells

Fig. 5. Example 5.3. The numerical solution (symbols "∘") and exact solution (solid lines) of density ρ, velocity v_1, and tangential velocity v_2 along the line $y = 0$. The PCP limited cells along $y = 0$ over time $t \in [0, 0.4]$ are also displayed.

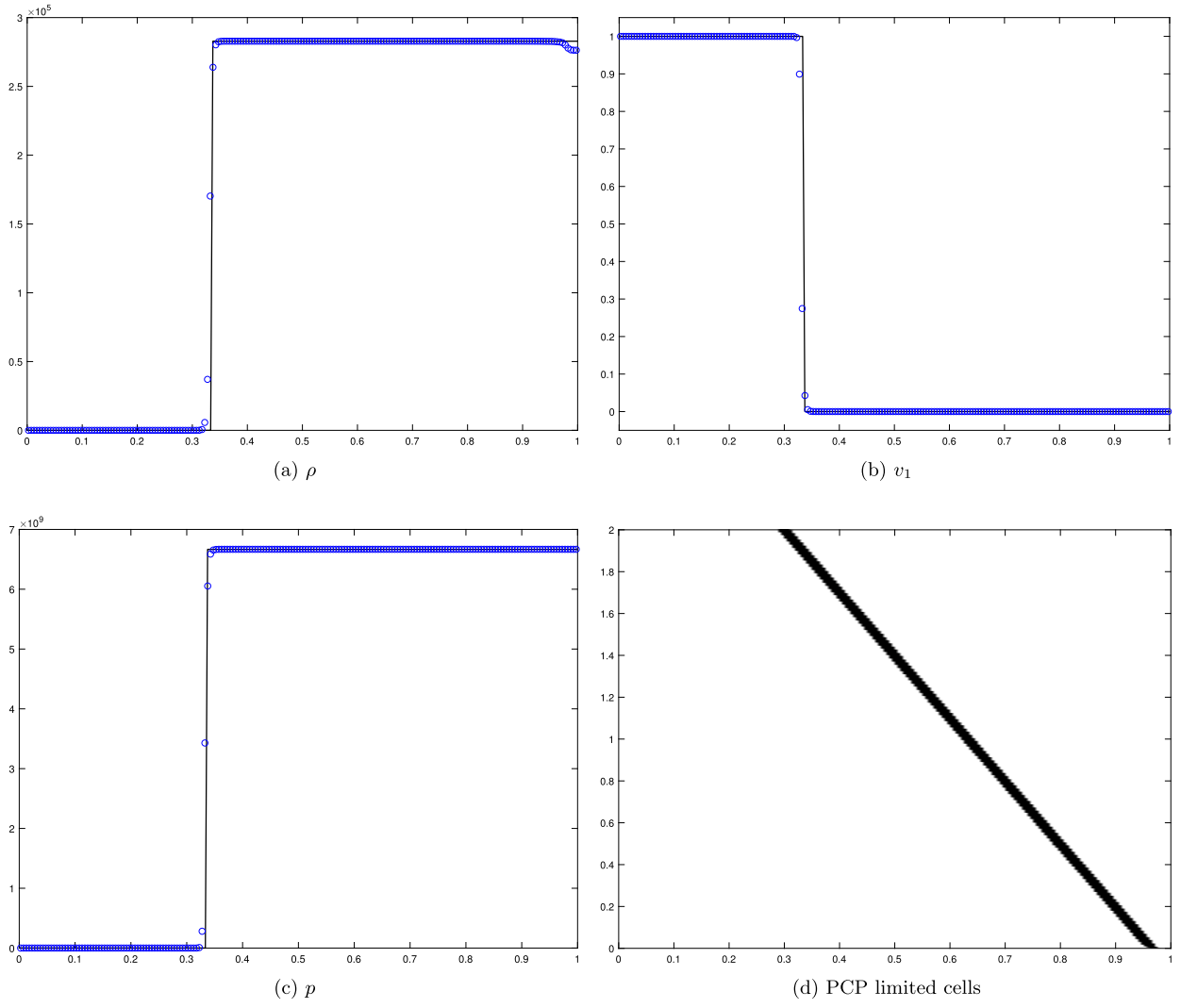


Fig. 6. Example 5.5. The numerical solution (symbols "∘") and exact solution (solid lines) of density ρ, velocity v_1, and pressure p at $t = 2$ as well as the PCP limited cells over time.

$$\boldsymbol{Q}(x,0) = \left(1, 1-10^{-10}, \frac{10^{-4}}{3}\right)^{\top}, \tag{5.6}$$

and the adiabatic index is take as $\gamma = 4/3$. The proposed model involves a scenario in which a gas with rightward velocity close to the speed of light collides with a wall. Upon impact, the kinetic energy of the gas is converted into internal energy, resulting in compression and heating. As a result of this process, a strong shock wave is generated, which propagates towards the left at a velocity of $v_s = (\gamma-1)W_0|v_0|/(W_0+1)$. Here, $v_0 = 1-10^{-10}$ represents the initial velocity of the gas, while W_0 denotes the corresponding Lorentz factor. The gas behind the shock wave comes to a rest and possesses a specific internal energy of $W_0 - 1$, as deduced through energy conservation across the wave. The compression ratio across the shock is given by $\sigma = (\gamma+1)/(\gamma-1) + (\gamma/(\gamma-1))(W_0-1)$.

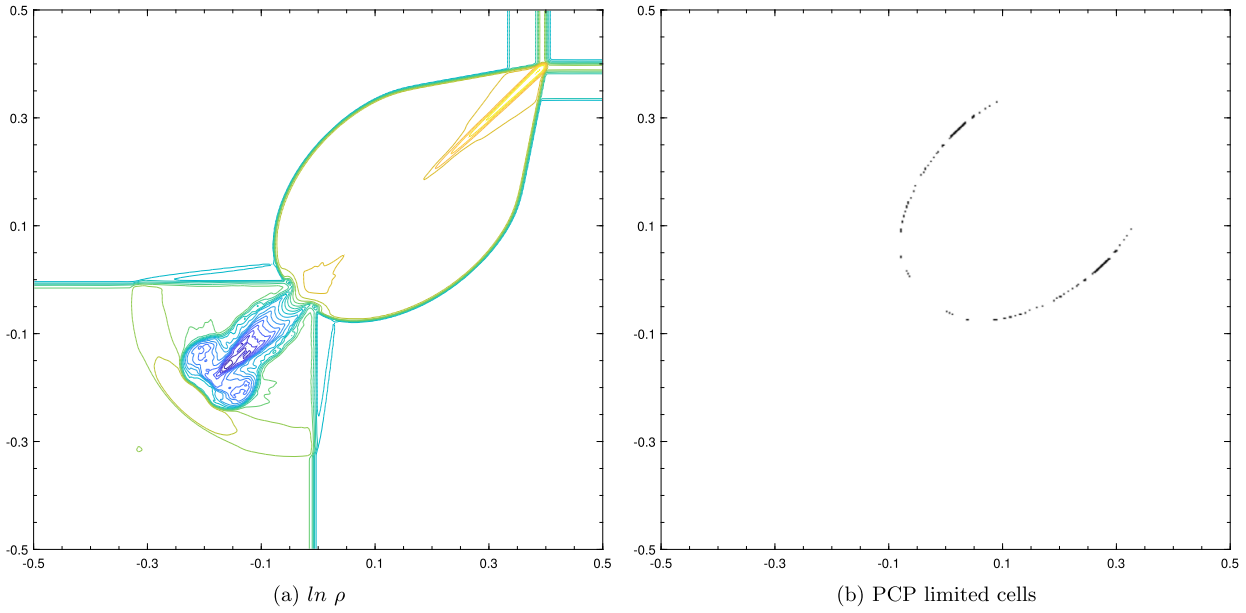
To evaluate the necessity of the PCP limiter, we perform the simulation without using this limiter and observe that the code breaks down after only one time step. We then apply the PCP limiter and plot the results at time $t = 2$ in Fig. 6, which also displays the cells where the PCP limiter is activated from $t = 0$ to 2. We observe that the PCP limiter is only activated in a few cells near the moving shock.

5.3. Two-dimensional examples

Example 5.6 *(2D smooth problem).* This example considers a 2D smooth problem in the domain $\Omega = [0, 2/\sqrt{3}] \times [0, 2]$ with the initial data

Table 5
Example 5.6: Numerical errors in L^1, L^2 and L^∞ norms and the corresponding convergence rates.

N	L^1 error	Order	L^2 error	Order	L^∞ error	Order	percentage of PCP limited cells
30	1.44E-03	-	4.14E-03	-	2.46E-02	-	12.72%
60	4.53E-06	8.31	1.79E-05	7.86	1.30E-04	7.56	0.59%
90	1.18E-10	26.02	1.32E-10	29.15	1.86E-10	33.19	0%
120	2.54E-11	5.35	2.83E-11	5.35	4.00E-11	5.34	0%
150	7.97E-12	5.20	8.85E-12	5.20	1.27E-11	5.14	0%
180	3.16E-12	5.08	3.51E-12	5.08	5.25E-12	4.85	0%



(a) $ln\ \rho$ (b) PCP limited cells

Fig. 7. Example 5.7. The contours of the density logarithm $\ln \rho$ with 25 equally spaced contour lines from -6 to 1.9 within the domain $[0,1]^2$ and the PCP limited cells at $t = 0.4$.

$$\boldsymbol{Q} = \left(1 + 0.999\sin[2\pi(x\cos\alpha + y\sin\alpha)], \frac{0.9}{\sqrt{2}}, \frac{0.9}{\sqrt{2}}, 0.01\right)^{\mathsf{T}}, \tag{5.7}$$

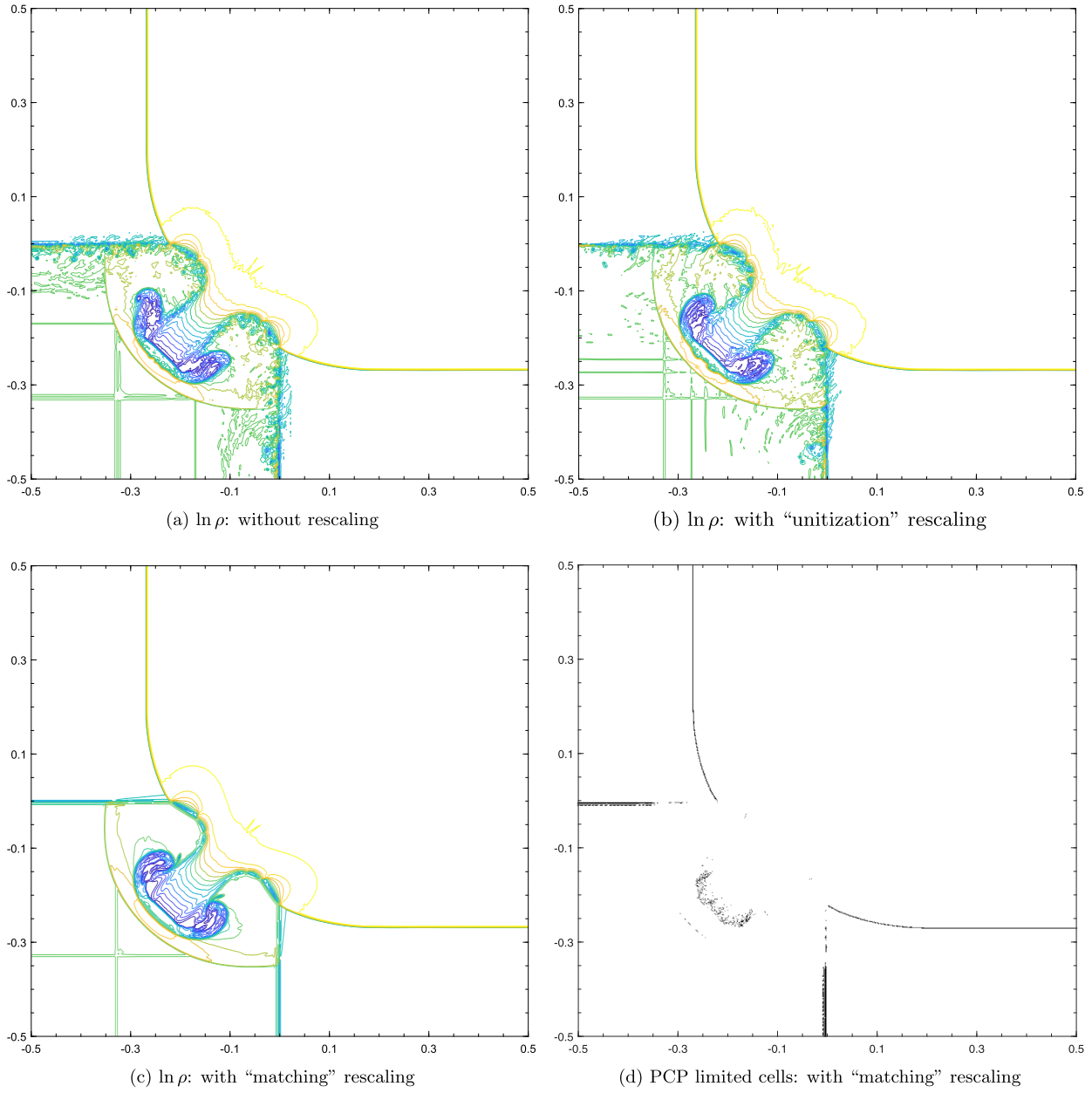
where $\alpha = \pi/6$. Due to the low density, large velocity close to the speed of light, and low pressure, the PCP limiting produce is necessary for successful simulation of this problem. The simulations are performed on the meshes of $N \times N$ uniform cells with varied $N \in \{30, 60, 90, \ldots, 180\}$. Table 5 lists the numerical errors of the mass density D and the convergence rates in L^1, L^2 and L^∞ norms at time $t = 0.05$. The results indicate that the 2D PCP HWENO scheme achieves fifth-order accuracy, which is not destroyed by the PCP limiter.

Example 5.7 *(2D Riemann problem I).* The use of 2D Riemann problems as benchmark tests has become widespread to evaluate the ability of a scheme to capture complex 2D relativistic wave configurations. Both this and the next tests simulate 2D Riemann problems of the ideal relativistic fluid within the domain $[-0.5, 0.5]^2$, which is divided into 400×400 uniform cells.

The initial conditions for this test are defined as follows:

$$\boldsymbol{Q}(x,y,0) = \begin{cases} (0.1,0,0,0.01)^{\mathsf{T}}, & x > 0,\ y > 0, \\ (0.1,0.99,0,1)^{\mathsf{T}}, & x < 0,\ y > 0, \\ (0.5,0,0,1)^{\mathsf{T}}, & x < 0,\ y < 0, \\ (0.1,0,0.99,1)^{\mathsf{T}}, & x > 0,\ y < 0. \end{cases}$$

Fig. 7 gives the contour of the density logarithm $\ln \rho$ and the cells where the PCP limiter is activated at $t = 0.4$. It is shown in the figure that the initial discontinuities in the four regions cause two reflected curved shock waves and a complex mushroom structure. The details in structure are consistent with those reported in previous works [50,4]. We observe that there are only a few PCP limited cells near the two reflected curved shocks.



(a) $\ln \rho$: without rescaling

(b) $\ln \rho$: with "unitization" rescaling

(c) $\ln \rho$: with "matching" rescaling

(d) PCP limited cells: with "matching" rescaling

Fig. 8. Example 5.8: The contours of the density logarithm $\ln \rho$ and the PCP limited cells at $t = 0.4$. Twenty-five equally spaced contour lines from -8 to 1.3.

Example 5.8 *(2D Riemann problem II)*. This example investigates a more ultra-relativistic 2D Riemann problem, which was first proposed in [50]. The initial conditions are defined as

$$\boldsymbol{Q}(x,y,0) = \begin{cases} (0.1,0,0,20)^{\mathsf{T}}, & x>0, y>0, \\ (0.00414329639576, 0.9946418833556542, 0, 0.05)^{\mathsf{T}}, & x<0, y>0, \\ (0.01,0,0,0.05)^{\mathsf{T}}, & x<0, y<0, \\ (0.00414329639576, 0, 0.9946418833556542, 0.05)^{\mathsf{T}}, & x>0, y<0. \end{cases}$$

In this problem, the maximum initial velocity of fluid is larger than that in 2D Riemann problem I. We use the proposed PCP HWENO scheme to simulate the problem on a mesh of 800×800 uniform cells. Furthermore, we utilize this problem to demonstrate the importance of rescaling eigenvectors in characteristic decomposition. We compare two rescaling approaches discussed in Remark 3.3.

Fig. 8 shows the contours of the density logarithm $\ln \rho$ at $t = 0.4$, obtained by our PCP HWENO scheme using three different methods: the "unitization" rescaling approach, the "matching" rescaling approach, and no rescaling. As we can see, the "matching"

rescaling approach exhibits the best performance, while the numerical solutions computed using the "unitization" rescaling approach and without rescaling exhibit serious oscillations.

Example 5.9 *(Shock-vortex interaction problems).* This example studies the interaction of a vortex with a shock. Pao and Salas [30] were the first to show this problem computationally in the non-relativistic case, while the special RHD case was studied in [1,7]. In our case, we have set the velocity magnitude of the vortex as $w = 0.9$ and the adiabatic index γ as 1.4. The initial rest-mass density and pressure are given by

$$\rho(x, y) = (1 - \alpha e^{1-r^2})^{\frac{1}{\gamma-1}}, \quad p = \rho^{\gamma}$$

where

$$\alpha = \frac{(\gamma - 1)/\gamma}{8\pi^2}\epsilon^2, \quad r = \sqrt{x_0^2 + y_0^2},$$

and ϵ represents the vortex strength. Using the Lorentz transformation, we can deduce that

$$x_0 = xW_w, \quad y_0 = y, \quad W_w = \frac{1}{\sqrt{1 - w^2}}.$$

The initial velocities are given by

$$v_1 = \frac{v_1^0 - w}{1 - v_1^0 w}, \qquad v_2 = \frac{v_2^0}{W_w(1 - v_1^0 w)},$$

where

$$(v_1^0, v_2^0) = (-y_0, x_0)f, \quad f = \sqrt{\frac{\beta}{1 + \beta r^2}}, \quad \beta = \frac{2\gamma\alpha e^{1-r^2}}{2\gamma - 1 - \gamma\alpha e^{1-r^2}}.$$

The computation domain is $[-17, 3] \times [-5, 5]$, which is divided into 800×400 uniform cells. The initial vortex is centered at $(0, 0)$, and there is a shock at $x = -6$ that far away from the vortex. The initial data in $x > 6$ can be calculated by the vortex condition above, and the post-shock state in $x < 6$ is given by

$$\boldsymbol{Q}(x, y, 0) = (4.891497310766981, -0.388882958251919, 0, 11.894863258311670)^{\top}.$$

We apply inflow and outflow boundary conditions at the right and left boundaries of the domain, respectively, and reflection boundary conditions are applied on the bottom and top boundaries.

We test our scheme in two different vortex strengths:

- A mild vortex with $\epsilon = 5$ as in [7].
- A demanding vortex with $\epsilon = 10.0828$ as in [4]. In this case, the minimum rest-mass density and pressure are 7.8337×10^{-15} and 1.7847×10^{-20}, respectively. We observe that the HWENO code without the PCP limiter cannot run this challenging test for even one time step, demonstrating the importance of the PCP limiter.

Figs. 9 and 10 show the schlieren images of $\log_{10}(1 + |\nabla\rho|)$ and $|\nabla p|$. The subtle structures in our results are in good agreement with those reported in [7,4], validating the effectiveness of our proposed PCP HWENO schemes in capturing complex waves and shocks.

Example 5.10 *(Axisymmetric relativistic jets).* This test simulates a relativistic jet by solving the RHD equations in cylindrical coordinates (see Section 4.3 for details). Relativistic jet flows have been extensively investigated by many researchers [27,50,57,31,4]. The computational domain consists of a 2D cylindrical box with dimensions of $(0 \le r \le 15,\ 0 \le z \le 45)$, which is discretized into 375×1125 uniform cells. The initial conditions in the computational domain are given by

$$\boldsymbol{Q}(r, z, 0) = (1, 0, 0, 1.70303 \times 10^{-4})^{\top}.$$

The relativistic jet beam has a velocity $v_b = 0.99$, density $\rho_b = 0.01$, and pressure $p_b = 1.70303 \times 10^{-4}$. The jet is injected through the inlet part ($r \le 1$) of the low-z boundary. At the symmetric axis $r = 0$, we use reflection conditions, and at the other parts of the boundaries, we impose outflow boundary conditions. It is worth noting that simulating such jets successfully is challenging because they typically involve ultra-relativistic regions, strong relativistic shock waves, shear flows, and interface instabilities.

We simulate this problem utilizing the PCP HWENO scheme combined with seven different primitive-variables-recovery algorithms to compare their efficiency and robustness. Table 6 summarizes the execution outcomes (either success or failure), total simulation time, and the time each algorithm uses for recovering primitive variables during the simulation. Due to the absence of both theoretical and empirical guidance on selecting a good initial value for the MPB-NR method [29], we adopt the pressure p from the last prior time step in the same cells as our initial value, as suggested by [41]. It is observed that the MPB-NR method fails to converge when the simulation reaches $t = 0.472757$, while all other algorithms successfully recover the primitive variables throughout the simulation. This indicates that, even with the incorporation of the PCP limiter, a robust algorithm for recovering primitive

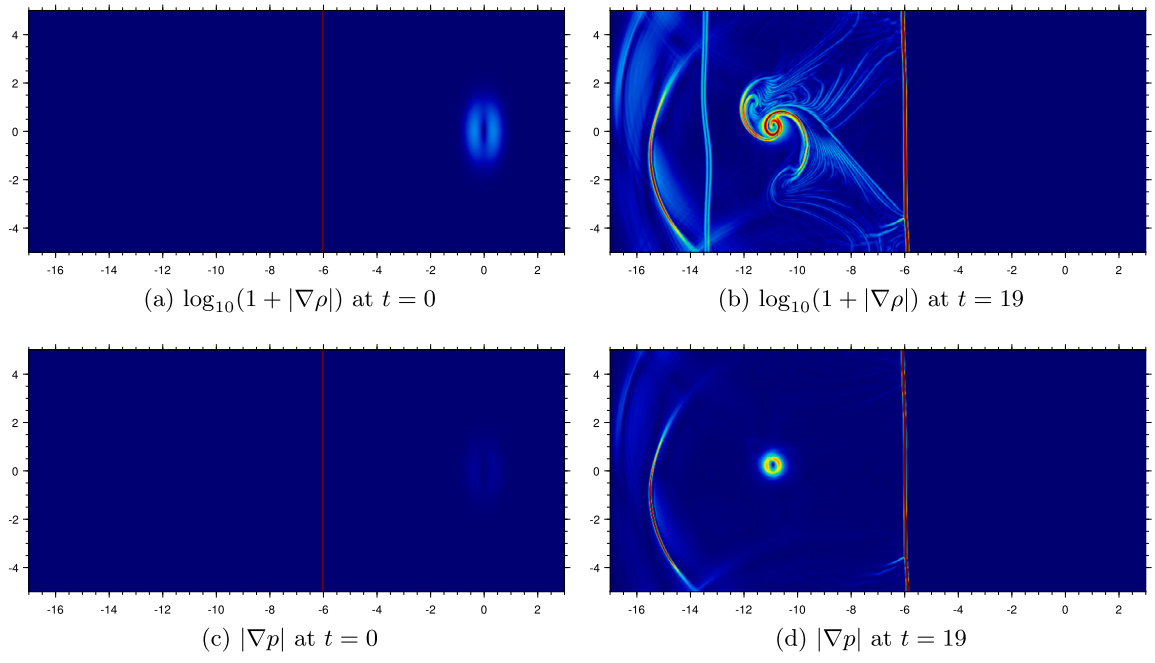


Fig. 9. Example 5.9 with the mild vortex: The schlieren images of $\log_{10}(1+|\nabla\rho|)$ from 0 to 1 and schlieren images of $|\nabla p|$ from 0 to 20.

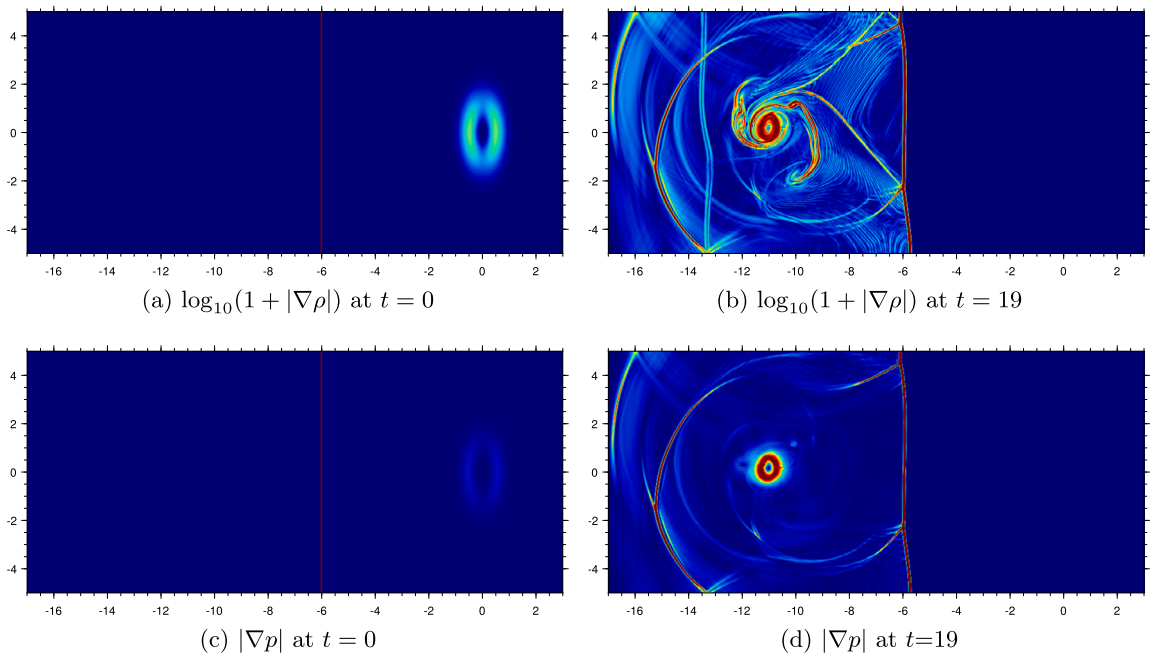


Fig. 10. Example 5.9 with the demanding vortex: the schlieren images of $\log_{10}(1+|\nabla\rho|)$ from 0 to 1 and schlieren images of $|\nabla p|$ from 0 to 20.

variables remains crucial to fully guarantee the PCP property. Otherwise, the computations would still fail due to the recovery of nonphysical primitive variables. Table 6 also reveals that among the six successful algorithms, the analytical algorithm exhibits the lowest computational efficiency. In contrast, the hybrid NR and NR-I methods stand out in terms of efficiency, reducing CPU time by approximately 10% to 40%. Consequently, our NR-I and Hybrid NR algorithms demonstrate high efficiency and stability in this test. Given the consistent outperformance of the hybrid NR method in accuracy, as demonstrated in Table 2, we recommend employing the proposed hybrid NR approach for recovering primitive variables.

Fig. 11 presents the schlieren images of the density logarithm $\ln\rho$ along with the PCP limited cells, obtained using our PCP HWENO scheme with the hybrid NR method at $t=60,80,100$. The images clearly demonstrate the formation of a bow shock by

Table 6
Performance of different primitive recovery algorithms for Example 5.10: Execution status, total simulation time, and primitive variables recovery time.

	NR-I	NR-II	Hybrid NR	Hybrid-linear	Analytical	Vel-Proxy	MPB-NR
Status	Success	Success	Success	Success	Success	Success	Failure
Total time	95h59m	99h39m	96h1m	125h34m	157h3m	104h44m	–
Recovery time	37h16m	41h15m	37h33m	67h13m	98h10m	46h23m	–

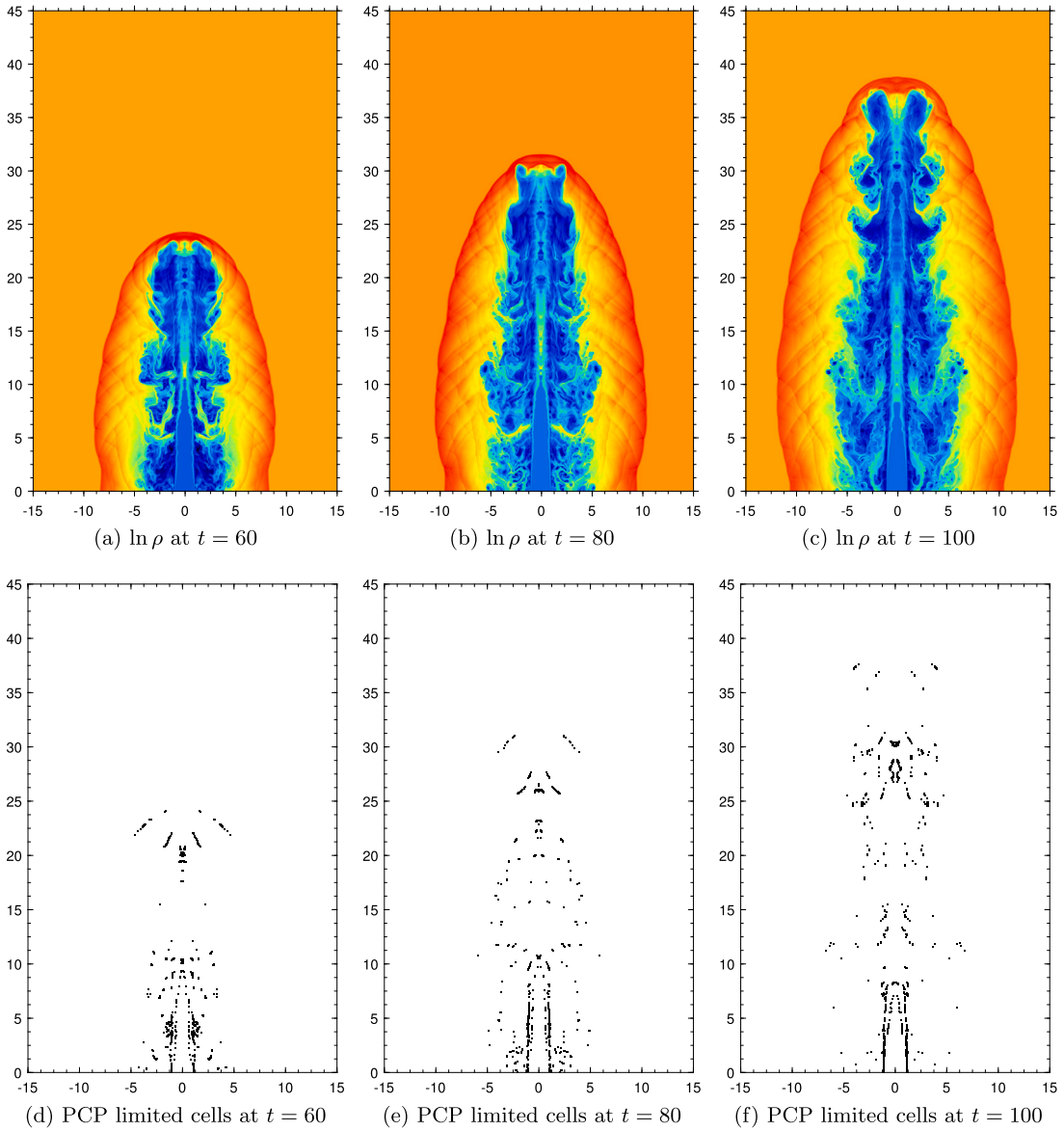


Fig. 11. Example 5.10: The schlieren images of the density logarithm $\ln\rho$ and the PCP limited cells.

the moving jet. The discontinuity between the jet material and the initial static material gives rise to Kelvin–Helmholtz instabilities. These results agree well with those reported in previous studies [27,50,57,4]. One can see that the PCP limiter is necessary in this challenging test, while only a small portion of cells is limited during the simulations.

6. Conclusion

Designing genuinely PCP schemes is a challenging task, as relativistic effects make it difficult to reformulate primitive variables explicitly using conservative variables. This paper proposed three efficient NR methods for robustly recovering primitive variables from conservative variables, and we exemplified their applications to develop PCP finite volume HWENO schemes for relativistic hydrodynamics. Our rigorous analysis demonstrated that these NR methods are always convergent and PCP, meaning that they preserve the physical constraints throughout the NR iterations. The discovery of these robust NR methods and their convergence analysis are very nontrivial. The presented PCP HWENO schemes were built on the NR methods, high-order HWENO reconstruction, a PCP limiter, and strong-stability-preserving time discretization. We rigorously demonstrated the PCP property of our schemes using convex decomposition techniques. In addition, we proposed the characteristic decomposition approach with rescaled eigenvectors and scale-invariant nonlinear weights to improve the performance of the HWENO schemes in simulating large-scale RHD problems. Several challenging numerical examples were provided to evaluate the robustness, accuracy, and high resolution of our PCP HWENO schemes and to demonstrate the efficiency of the proposed NR methods.

Indeed, the proposed NR methods are versatile and can be integrated with any RHD schemes requiring the recovery of primitive variables. By stating that our three convergent NR methods are "efficient", we are *not* implying that the other methods are "inefficient". In fact, based on our numerical tests, the Vel-Proxy method [37] is also an efficient approach, but there is yet no rigorous proof for its convergence and PCP property.

CRediT authorship contribution statement

- **Chaoyi Cai:** Conceptualization, Methodology, Code, Software, Investigation, Visualization, Writing - Original Draft;
- **Jianxian Qiu:** Supervision, Conceptualization, Methodology, Validation, Resources, Writing - Review & Editing;
- **Kailiang Wu:** Conceptualization, Methodology, Supervision, Validation, Resources, Writing - Review & Editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix A. 2D Linear reconstruction operator M_L

For convenience, we number the night cells around the cell $I_{i,j}$, as shown in Fig. 3. We denote $\xi(x)=\frac{x-x_i}{\Delta x}$, $\eta(y)=\frac{y-y_j}{\Delta y}$. Let us reconstruct a quartic polynomial $P_0(x,y)=\sum_{s=0}^{4-r}\sum_{r=0}^{4} a_{s,r}^0\xi(x)^s\eta(y)^r$ satisfying

$$\begin{cases} \dfrac{1}{\Delta x\Delta y}\displaystyle\int_{I_{i,j}^k} P_0(x,y)dxdy=u_k, \quad k=1,\dots,9,\\ \dfrac{1}{\Delta x\Delta y}\displaystyle\int_{I_{i,j}^5} P_0(x,y)\frac{x-x_i}{\Delta x}dxdy=v_5,\\ \dfrac{1}{\Delta x\Delta y}\displaystyle\int_{I_{i,j}^5} P_0(x,y)\frac{y-y_j}{\Delta y}dxdy=w_5 \end{cases} \tag{A.1}$$

and minimizing

$$\sqrt{\sum_{k=2,4,6,8}\left[\left(\frac{1}{\Delta x\Delta y}\int_{I_{i,j}^k} P_0(x,y)\frac{x-x_i}{\Delta x}dxdy-v_k\right)^2+\left(\frac{1}{\Delta x\Delta y}\int_{I_{i,j}^k} P_0(x,y)\frac{y-y_j}{\Delta y}dxdy-w_k\right)^2\right]}, \tag{A.2}$$

where u_k, v_k, w_k are any given real numbers. The conditions (A.1)–(A.2) form a constrained least squares problem for the unknowns $\left\{a_{s,r}^0\right\}_{s,r=0}^{4}$. Solving this problem with the nullspace method (see [2, Section 5.1.3] for details) gives the expressions of $\{a_{s,r}^0\}$, which are the linear combinations of

$$\begin{bmatrix} u_7 & u_8 & u_9 \\ u_4 & u_5 & u_6 \\ u_1 & u_2 & u_3 \end{bmatrix}, \begin{bmatrix} & v_8 & \\ v_4 & v_5 & v_6 \\ & v_2 & \end{bmatrix}, \text{ and } \begin{bmatrix} & w_8 & \\ w_4 & w_5 & w_6 \\ & w_2 & \end{bmatrix}.$$

In order to save space, we here omit the specific expressions of $a^0_{s,r}$. Define the operator

$$M_L([u_1,\dots,u_9],[v_2\ v_4\ v_5\ v_6\ v_8],[w_2\ w_4\ w_5\ w_6\ w_8],\xi,\eta) := P_0(x(\xi),y(\eta)) = \sum_{s=0}^{4-r}\sum_{r=0}^{4} a^0_{s,r}\xi^s\eta^r$$

which is a mapping from $\mathbb{R}^{1\times 9}\times\mathbb{R}^{1\times 5}\times\mathbb{R}^{1\times 5}\times\mathbb{R}\times\mathbb{R}$ to $\mathbb{R}$. Using this operator, it is convenient to compute the value of $P_0(x,y)$ at $(x_{i+\xi},y_{j+\eta})$ with

$$P_0(x_{i+\xi},y_{j+\eta}) = M_L([u_1,\dots,u_9],[v_2\ v_4\ v_5\ v_6\ v_8],[w_2\ w_4\ w_5\ w_6\ w_8],\xi,\eta).$$

The operator M_L represents the reconstruction mapping for the scalar equation. In order to extend the reconstruction to the 2D RHD equations, we generalize it to vector cases component-wisely as follows

$$\begin{aligned}&\boldsymbol{M}_L\left(\left[\boldsymbol{U}_1,\dots,\boldsymbol{U}_9\right],\left[\boldsymbol{V}_2\ \boldsymbol{V}_4\ \boldsymbol{V}_5\ \boldsymbol{V}_6\ \boldsymbol{V}_8\right],\left[\boldsymbol{W}_2\ \boldsymbol{W}_4\ \boldsymbol{W}_5\ \boldsymbol{W}_6\ \boldsymbol{W}_8\right],\xi,\eta\right)\\ :=&\begin{pmatrix} M_L\left(\left[U_1^{(1)},\dots,U_9^{(1)}\right],\left[V_2^{(1)}\ V_4^{(1)}\ V_5^{(1)}\ V_6^{(1)}\ V_8^{(1)}\right],\left[W_2^{(1)}\ W_4^{(1)}\ W_5^{(1)}\ W_6^{(1)}\ W_8^{(1)}\right],\xi,\eta\right)\\ M_L\left(\left[U_1^{(2)},\dots,U_9^{(2)}\right],\left[V_2^{(2)}\ V_4^{(2)}\ V_5^{(2)}\ V_6^{(2)}\ V_8^{(2)}\right],\left[W_2^{(2)}\ W_4^{(2)}\ W_5^{(2)}\ W_6^{(2)}\ W_8^{(2)}\right],\xi,\eta\right)\\ M_L\left(\left[U_1^{(3)},\dots,U_9^{(3)}\right],\left[V_2^{(3)}\ V_4^{(3)}\ V_5^{(3)}\ V_6^{(3)}\ V_8^{(3)}\right],\left[W_2^{(3)}\ W_4^{(3)}\ W_5^{(3)}\ W_6^{(3)}\ W_8^{(3)}\right],\xi,\eta\right)\\ M_L\left(\left[U_1^{(4)},\dots,U_9^{(4)}\right],\left[V_2^{(4)}\ V_4^{(4)}\ V_5^{(4)}\ V_6^{(4)}\ V_8^{(4)}\right],\left[W_2^{(4)}\ W_4^{(4)}\ W_5^{(4)}\ W_6^{(4)}\ W_8^{(4)}\right],\xi,\eta\right)\end{pmatrix},\end{aligned}$$

where $\boldsymbol{M}_L$ is the reconstruction operator from $\mathbb{R}^{4\times 9}\times\mathbb{R}^{4\times 5}\times\mathbb{R}^{4\times 5}\times\mathbb{R}\times\mathbb{R}$ to $\mathbb{R}^{4\times 1}$. It is worth pointing out that $\boldsymbol{M}_L(\cdot,\cdot,\cdot,\xi,\eta)$ is a linear mapping for fixed ξ and η,

Appendix B. 2D HWENO reconstruction operator $\boldsymbol{M}_H$

Reconstruct four quadratic polynomials $P_n(x,y) := \sum_{s=0}^{2-r}\sum_{r=0}^{2} a^r_{s,r}\xi(x)^s\eta(y)^r$ $(r=1,2,3,4)$ satisfying

$$\begin{cases} \dfrac{1}{\Delta x\Delta y}\displaystyle\int_{I^k_{i,j}} P_1(x,y)dxdy = u_k, \quad k=1,2,4,5,\\ \dfrac{1}{\Delta x\Delta y}\displaystyle\int_{I^k_{i,j}} P_2(x,y)dxdy = u_k, \quad k=2,3,5,6,\\ \dfrac{1}{\Delta x\Delta y}\displaystyle\int_{I^k_{i,j}} P_3(x,y)dxdy = u_k, \quad k=4,5,7,8,\\ \dfrac{1}{\Delta x\Delta y}\displaystyle\int_{I^k_{i,j}} P_4(x,y)dxdy = u_k, \quad k=5,6,8,9,\\ \dfrac{1}{\Delta x\Delta y}\displaystyle\int_{I^5_{i,j}} P_n(x,y)\frac{x-x_i}{\Delta x}dxdy = v_5, \quad n=1,2,3,4,\\ \dfrac{1}{\Delta x\Delta y}\displaystyle\int_{I^5_{i,j}} P_n(x,y)\frac{y-y_j}{\Delta y}dxdy = w_5, \quad n=1,2,3,4. \end{cases} \tag{B.1}$$

Similarly, we can obtain the expressions of $a^n_{s,r}$ $(n=1,2,3,4,\ s,r=1,2)$, which are linear combinations of

$$\begin{bmatrix} u_7 & u_8 & u_9 \\ u_4 & u_5 & u_6 \\ u_1 & u_2 & u_3 \end{bmatrix},\ v_5,\ \text{and}\ w_5.$$

Next, in order to measure the smoothness of the polynomial $P_n(x,y)$ in the cell $I_{i,j}$, we calculate the smooth indicators, with the same definition as in [62],

$$\beta_n = \sum_{|l|=1}^{r} |I_{i,j}|^{|l|-1}\int_{I_i,j}\left(\frac{\partial^{|l|}}{\partial x^{l_1}\partial y^{l_2}}P_n(x,y)\right)^2 \mathrm{d}x\mathrm{d}y,\ n=0,\dots,4,$$

where r is the degree of the polynomials $P_n(x,y)$. Then the HWENO reconstruction polynomial is defined by

$$P_H(x,y)=\omega_0\left(\frac{1}{\gamma_0}P_0(x,y)-\sum_{n=1}^{4}\frac{\gamma_n}{\gamma_0}P_n(x,y)\right)+\sum_{n=1}^{4}\omega_n P_n(x,y),$$

where the nonlinear weights

$$\omega_n=\frac{\bar{\omega}_n}{\sum_{k=0}^{4}\bar{\omega}_k}\quad\text{with}\quad\bar{\omega}_n=\gamma_n\left(1+\frac{\tau^2}{(\beta_n)^2+\epsilon}\right),n=0,\dots,4,\tag{B.2}$$

and $\tau:=\left(\frac{\sum_{n=0}^{4}|\beta_0-\beta_n|}{4}\right)$. Similar to the 1D case, these nonlinear weights possess the "scaling-invariant" property.

Define the operator

$$M_H([u_1,\dots,u_9],[v_2\ v_4\ v_5\ v_6\ v_8],[w_2\ w_4\ w_5\ w_6\ w_8],\xi,\eta):=P_H(x(\xi),y(\eta)),$$

which is a mapping from $\mathbb{R}^{1\times9}\times\mathbb{R}^{1\times5}\times\mathbb{R}^{1\times5}\times\mathbb{R}\times\mathbb{R}$ to $\mathbb{R}$. It is easy to compute the value of $P_H(x,y)$ at $(x_{i+\xi},y_{j+\eta})$ with

$$P_H(x_{i+\xi},y_{j+\eta})=M_H([u_1,\dots,u_9],[v_2\ v_4\ v_5\ v_6\ v_8],[w_2\ w_4\ w_5\ w_6\ w_8],\xi,\eta).$$

We can generalize the scalar HWENO reconstruction operator M_H to the vector cases in a component by component manner:

$$\begin{aligned}&\boldsymbol{M}_H\left(\left[\boldsymbol{U}_1,\dots,\boldsymbol{U}_9\right],\left[\boldsymbol{V}_2\ \boldsymbol{V}_4\ \boldsymbol{V}_5\ \boldsymbol{V}_6\ \boldsymbol{V}_8\right],\left[\boldsymbol{W}_2\ \boldsymbol{W}_4\ \boldsymbol{W}_5\ \boldsymbol{W}_6\ \boldsymbol{W}_8\right],\xi,\eta\right)\\&:=\begin{pmatrix}M_H\left(\left[U_1^{(1)},\dots,U_9^{(1)}\right],\left[V_2^{(1)}\ V_4^{(1)}\ V_5^{(1)}\ V_6^{(1)}\ V_8^{(1)}\right],\left[W_2^{(1)}\ W_4^{(1)}\ W_5^{(1)}\ W_6^{(1)}\ W_8^{(1)}\right],\xi,\eta\right)\\M_H\left(\left[U_1^{(2)},\dots,U_9^{(2)}\right],\left[V_2^{(2)}\ V_4^{(2)}\ V_5^{(2)}\ V_6^{(2)}\ V_8^{(2)}\right],\left[W_2^{(2)}\ W_4^{(2)}\ W_5^{(2)}\ W_6^{(2)}\ W_8^{(2)}\right],\xi,\eta\right)\\M_H\left(\left[U_1^{(3)},\dots,U_9^{(3)}\right],\left[V_2^{(3)}\ V_4^{(3)}\ V_5^{(3)}\ V_6^{(3)}\ V_8^{(3)}\right],\left[W_2^{(3)}\ W_4^{(3)}\ W_5^{(3)}\ W_6^{(3)}\ W_8^{(3)}\right],\xi,\eta\right)\\M_H\left(\left[U_1^{(4)},\dots,U_9^{(4)}\right],\left[V_2^{(4)}\ V_4^{(4)}\ V_5^{(4)}\ V_6^{(4)}\ V_8^{(4)}\right],\left[W_2^{(4)}\ W_4^{(4)}\ W_5^{(4)}\ W_6^{(4)}\ W_8^{(4)}\right],\xi,\eta\right)\end{pmatrix},\end{aligned}$$

where $U_{i,j}^{(\ell)}$ is the ℓth component of $\boldsymbol{U}_{i,j}$, $V_{i,j}^{(\ell)}$ is the ℓth component of $\boldsymbol{V}_{i,j}$, $W_{i,j}^{(\ell)}$ is the ℓth component of $\boldsymbol{W}_{i,j}$. Different from $\boldsymbol{M}_L$, the operator $\boldsymbol{M}_H(\cdot,\dots,\cdot,\xi,\eta)$ is a nonlinear mapping for fixed ξ and η.

References

[1] D.S. Balsara, J. Kim, A subluminal relativistic magnetohydrodynamics scheme with ADER-WENO predictor and multidimensional Riemann solver-based corrector, J. Comput. Phys. 312 (2016) 357–384.
[2] Å. Björck, Numerical Methods for Least Squares Problems, Society for Industrial and Applied Mathematics, 1996.
[3] M. Castro, B. Costa, W.S. Don, High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws, J. Comput. Phys. 230 (5) (2011) 1766–1792.
[4] Y. Chen, K. Wu, A physical-constraint-preserving finite volume WENO method for special relativistic hydrodynamics on unstructured meshes, J. Comput. Phys. 466 (2022) 111398.
[5] B. Costa, W.S. Don, Multi-domain hybrid spectral-WENO methods for hyperbolic conservation laws, J. Comput. Phys. 224 (2) (2007) 970–991.
[6] A. Dolezal, S. Wong, Relativistic hydrodynamics and essentially non-oscillatory shock capturing schemes, J. Comput. Phys. 120 (2) (1995) 266–277.
[7] J. Duan, H. Tang, High-order accurate entropy stable finite difference schemes for one- and two-dimensional special relativistic hydrodynamics, Adv. Appl. Math. Mech. 12 (1) (2019) 1–29.
[8] D.K. Dunaway, B.L. Turlington, Some major modifications to a new method for solving ill-conditioned polynomial equations, in: Proceedings of the ACM Annual Conference, vol. 2, 1972, pp. 636–643.
[9] F. Eulderink, G. Mellema, General relativistic hydrodynamics with a Roe solver, Astron. Astrophys. Suppl. Ser. 110 (1995) 587.
[10] N. Flocke, Algorithm 954: an accurate and efficient cubic and quartic equation solver for physical applications, ACM Trans. Math. Softw. 41 (4) (2015) 1–24.
[11] J.A. Font, Numerical hydrodynamics and magnetohydrodynamics in general relativity, Living Rev. Relativ. 11 (1) (2008) 1–131.
[12] A. Harten, High resolution schemes for hyperbolic conservation laws, J. Comput. Phys. 49 (1983) 357–393.
[13] A. Harten, Preliminary results on the extension of ENO schemes to two-dimensional problems, in: Nonlinear Hyperbolic Problems, vol. 1270, 1987, pp. 23–40.
[14] A. Harten, B. Engquist, S. Osher, S.R. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes, III, J. Comput. Phys. 71 (2) (1987) 231–303.
[15] P. He, H. Tang, An adaptive moving mesh method for two-dimensional relativistic hydrodynamics, Commun. Comput. Phys. 11 (1) (2012) 114–146.
[16] C. Hu, C.-W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys. 150 (1) (1999) 97–127.
[17] X.Y. Hu, N.A. Adams, C.-W. Shu, Positivity-preserving method for high-order conservative schemes solving compressible Euler equations, J. Comput. Phys. 242 (2013) 169–180.
[18] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1) (1996) 202–228.
[19] L.E. Kidder, S.E. Field, F. Foucart, E. Schnetter, S.A. Teukolsky, A. Bohn, N. Deppe, P. Diener, F. Hébert, J. Lippuner, et al., SpECTRE: a task-based discontinuous Galerkin code for relativistic astrophysics, J. Comput. Phys. 335 (2017) 84–114.
[20] L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeon, J.E. Flaherty, Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws, Appl. Numer. Math. 48 (3–4) (2004) 323–338.
[21] D. Levy, G. Puppo, G. Russo, Central WENO schemes for hyperbolic systems of conservation laws, ESAIM: Math. Model. Numer. Anal. 33 (3) (1999) 547–571.
[22] G. Li, J. Qiu, Hybrid weighted essentially non-oscillatory schemes with different indicators, J. Comput. Phys. 229 (21) (2010) 8105–8129.
[23] D. Ling, J. Duan, H. Tang, Physical-constraints-preserving Lagrangian finite volume schemes for one- and two-dimensional special relativistic hydrodynamics, J. Comput. Phys. 396 (2019) 507–543.
[24] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1) (1994) 200–212.

[25] J.M. Martíand, E. Müller, Numerical hydrodynamics in special relativity, Living Rev. Relativ. 6 (1) (2003) 1–100.
[26] J.M. Martíand, E. Müller, Grid-based methods in relativistic hydrodynamics and magnetohydrodynamics, Living Rev. Comput. Astrophys. 1 (1) (2015) 1–182.
[27] J.M. Martí, E. Müller, J. Font, J.M.Z. Ibáñez, A. Marquina, Morphology and dynamics of relativistic jets, Astrophys. J. 479 (1) (1997) 151.
[28] A. Mignone, G. Bodo, An HLLC Riemann solver for relativistic flows—I. Hydrodynamics, Mon. Not. R. Astron. Soc. 364 (1) (2005) 126–136.
[29] A. Mignone, T. Plewa, G. Bodo, The piecewise parabolic method for multidimensional relativistic fluid dynamics, Astrophys. J. Suppl. Ser. 160 (1) (2005) 199.
[30] S. Pao, M. Salas, A numerical study of two-dimensional shock vortex interaction, in: 14th Fluid and Plasma Dynamics Conference, 1981, p. 1205.
[31] T. Qin, C.-W. Shu, Y. Yang, Bound-preserving discontinuous Galerkin methods for relativistic hydrodynamics, J. Comput. Phys. 315 (2016) 323–347.
[32] 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 (1) (2004) 115–135.
[33] 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 (6) (2005) 642–663.
[34] D. Radice, L. Rezzolla, Discontinuous Galerkin methods for general-relativistic hydrodynamics: formulation and application to spherically symmetric spacetimes, Phys. Rev. D 84 (2) (2011) 024010.
[35] D. Radice, L. Rezzolla, THC: a new high-order finite-difference high-resolution shock-capturing code for special-relativistic hydrodynamics, Astron. Astrophys. 547 (2012) A26.
[36] D. Radice, L. Rezzolla, F. Galeazzi, High-order fully general-relativistic hydrodynamics: new approaches and tests, Class. Quantum Gravity 31 (7) (2014) 075012.
[37] G. Riccardi, D. Durante, Primitive variable recovering in special relativistic hydrodynamics allowing ultra-relativistic flows, in: International Mathematical Forum, vol. 42, 2008, pp. 2081–2111.
[38] D. Ryu, I. Chattopadhyay, E. Choi, Equation of state in numerical relativistic hydrodynamics, Astrophys. J. Suppl. Ser. 166 (1) (2006) 410.
[39] J. Shi, C. Hu, C.-W. Shu, A technique of treating negative weights in WENO schemes, J. Comput. Phys. 175 (1) (2002) 108–127.
[40] C.-W. Shu, Bound-preserving high-order schemes for hyperbolic equations: survey and recent developments, in: XVI International Conference on Hyperbolic Problems: Theory, Numerics, Applications, 2016, pp. 591–603.
[41] D.M. Siegel, P. Mösta, D. Desai, S. Wu, Recovery schemes for primitive variables in general-relativistic magnetohydrodynamics, Astrophys. J. 859 (1) (2018) 71.
[42] A. Tchekhovskoy, J.C. McKinney, R. Narayan, WHAM: a WENO-based general relativistic numerical scheme–I. Hydrodynamics, Mon. Not. R. Astron. Soc. 379 (2) (2007) 469–497.
[43] S.A. Teukolsky, Formulation of discontinuous Galerkin methods for relativistic astrophysics, J. Comput. Phys. 312 (2016) 333–356.
[44] J.R. Wilson, G.J. Mathews, Relativistic Numerical Hydrodynamics, Cambridge University Press, 2003.
[45] K. Wu, Design of provably physical-constraint-preserving methods for general relativistic hydrodynamics, Phys. Rev. D 95 (10) (2017) 103001.
[46] K. Wu, Positivity-preserving analysis of numerical schemes for ideal magnetohydrodynamics, SIAM J. Numer. Anal. 56 (4) (2018) 2124–2147.
[47] K. Wu, Minimum principle on specific entropy and high-order accurate invariant-region-preserving numerical methods for relativistic hydrodynamics, SIAM J. Sci. Comput. 43 (6) (2021) B1164–B1197.
[48] K. Wu, C.-W. Shu, Provably physical-constraint-preserving discontinuous Galerkin methods for multidimensional relativistic MHD equations, Numer. Math. 148 (3) (2021) 699–741.
[49] K. Wu, C.-W. Shu, Geometric quasilinearization framework for analysis and design of bound-preserving schemes, SIAM Rev. 65 (4) (2023) 1031–1073.
[50] K. Wu, H. Tang, High-order accurate physical-constraints-preserving finite difference WENO schemes for special relativistic hydrodynamics, J. Comput. Phys. 298 (2015) 539–564.
[51] K. Wu, H. Tang, Physical-constraint-preserving central discontinuous Galerkin methods for special relativistic hydrodynamics with a general equation of state, Astrophys. J. Suppl. Ser. 228 (1) (2016) 3.
[52] K. Wu, H. Tang, Admissible states and physical-constraints-preserving schemes for relativistic magnetohydrodynamic equations, Math. Models Methods Appl. Sci. 27 (10) (2017) 1871–1928.
[53] T. Xiong, J.-M. Qiu, Z. Xu, Parametrized positivity preserving flux limiters for the high order finite difference WENO scheme solving compressible Euler equations, J. Sci. Comput. 67 (3) (2016) 1066–1088.
[54] Z. Xu, Parametrized maximum principle preserving flux limiters for high order schemes solving hyperbolic conservation laws: one-dimensional scalar problem, Math. Comput. 83 (289) (2014) 2213–2238.
[55] Z. Xu, X. Zhang, Bound-Preserving High-Order Schemes, Handbook of Numerical Analysis, vol. 18, Elsevier, 2017, pp. 81–102. Chapter 4.
[56] O. Zanotti, M. Dumbser, A high order special relativistic hydrodynamic and magnetohydrodynamic code with space–time adaptive mesh refinement, Comput. Phys. Commun. 188 (2015) 110–127.
[57] W. Zhang, A.I. MacFadyen RAM, A relativistic adaptive mesh refinement hydrodynamics code, Astrophys. J. Suppl. Ser. 164 (1) (2006) 255.
[58] X. Zhang, C.-W. Shu, On maximum-principle-satisfying high order schemes for scalar conservation laws, J. Comput. Phys. 229 (9) (2010) 3091–3120.
[59] X. Zhang, C.-W. Shu, On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes, J. Comput. Phys. 229 (23) (2010) 8918–8934.
[60] J. Zhao, H. Tang, Runge–Kutta discontinuous Galerkin methods with WENO limiter for the special relativistic hydrodynamics, J. Comput. Phys. 242 (2013) 138–168.
[61] Z. Zhao, Y. Chen, J. Qiu, A hybrid Hermite WENO scheme for hyperbolic conservation laws, J. Comput. Phys. 405 (2020) 109175.
[62] Z. Zhao, J. Qiu, A Hermite WENO scheme with artificial linear weights for hyperbolic conservation laws, J. Comput. Phys. 417 (2020) 109583.
[63] Z. Zhao, J. Zhu, Y. Chen, J. Qiu, A new hybrid WENO scheme for hyperbolic conservation laws, Comput. Fluids 179 (2019) 422–436.
[64] J. Zhu, J. Qiu, A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws, J. Comput. Phys. 318 (2016) 110–121.
[65] J. Zhu, J. Qiu, A new type of finite volume weno schemes for hyperbolic conservation laws, J. Sci. Comput. 73 (2017) 1338–1359.