Commun. Comput. Phys.
doi: 10.4208/cicp.291010.180311a

Vol. **11**, No. 1, pp. 114-146
January 2012

An Adaptive Moving Mesh Method for Two-Dimensional Relativistic Hydrodynamics

Peng He and Huazhong Tang*

*HEDPS, CAPT & LMAM, School of Mathematical Sciences, Peking University,
Beijing 100871, China.*

Received 29 October 2010; Accepted (in revised version) 18 March 2011

Available online 5 September 2011

Abstract. This paper extends the adaptive moving mesh method developed by Tang and Tang [36] to two-dimensional (2D) relativistic hydrodynamic (RHD) equations. The algorithm consists of two "independent" parts: the time evolution of the RHD equations and the (static) mesh iteration redistribution. In the first part, the RHD equations are discretized by using a high resolution finite volume scheme on the fixed but nonuniform meshes without the full characteristic decomposition of the governing equations. The second part is an iterative procedure. In each iteration, the mesh points are first redistributed, and then the cell averages of the conservative variables are remapped onto the new mesh in a conservative way. Several numerical examples are given to demonstrate the accuracy and effectiveness of the proposed method.

AMS subject classifications: 35L65, 65M50, 74S10, 76L05, 76N15, 76Y05

Key words: Adaptive moving mesh method, finite volume method, conservative interpolation, relativistic hydrodynamics.

1 Introduction

Relativistic hydrodynamics plays a major role in many fields of modern physics, e.g., astrophysics, nuclear and high-energy physics and, lately, also in condensed matter. A relativistic description of fluid dynamics should be used whenever matter is influenced by large gravitational potentials, where a description in terms of the Einstein field theory of gravity is necessary. It is also necessary in situations where the local velocity of the flow is close to the light speed in vacuum or where the local internal energy density is comparable (or larger) than the local rest mass density of the fluid. Alternatively, relativistic flows are present in numerous astrophysical phenomena from stellar to galactic

*Corresponding author. *Email addresses:* `hepengpku@gmail.com` (P. He), `hztang@math.pku.edu.cn` (H. Z. Tang)

scales, e.g., core collapse super-novae, X-ray binaries, pulsars, coalescing neutron stars and black holes, micro-quasars, active galactic nuclei, super-luminal jets, and gamma-ray bursts etc.

The dynamics of the relativistic systems require solving highly nonlinear equations, rendering the analytic treatment of practical problems extremely difficult. Often studying them numerically is a possible approach. The first attempt to solve the RHD equations was made by Wilson in the 1970s [42,43], using an Eulerian explicit finite difference code with a monotonic transport, depending on the artificial viscosity techniques to handle shock waves. After that, various numerical methods are developed to solve the RHD equations. We refer the readers to the review article by Martí and Müller [27]. Most of those schemes are the generalizations of the shock-capturing Godunov-type methods based on the exact or approximate Riemann solvers. These Riemann solvers either rely on characteristic decompositions of the Jacobian matrix or not. Eulderink and Mellema [15] and Falle and Komissarov [16] developed RHD solvers based on the local linearization, respectively. Balsara [1], Dai and Woodward [8], and Mignone et al. [29] developed two-shock approximation solver for the RHD system. A flux-splitting method was extended to the RHDs in [12]. Schneider et al. [32] and Duncan and Hughes [13] presented the HLL (Harten-Lax-van Leer) method in the context of the RHD equations. An extension of the HLLC (Harten-Lax-van Leer-Contact) approximate Riemann solver for the RHDs was presented by Mignone and Bodo [28]. ENO (essentially non-oscillatory) based methods for the RHD system have been studied by Dolezal and Wong [11] and Del Zanna and Bucciantini [45].

In practice, solutions to the (nonlinear) RHD equations are frequently smooth in large fractions of the physical domain but contain sharp transitions or discontinuities in relatively localized regions. In the smooth regions, relatively coarse numerical zoning may be sufficient to accurately represent the solution, while finer zoning is needed where sharp transitions occur. Because of this, adaptive mesh strategies are needed. Successful implementation of the adaptive approaches can improve the accuracy of the numerical approximation and decrease the computational cost. Adaptive moving mesh methods have been playing an increasingly important role in many branches of scientific and engineering areas. Up to now, there have been many important progresses in the adaptive moving mesh methods for partial differential equations, including grid redistribution approaches based on the variational principle of Winslow [44], Brackbill [2], Brackbill and Saltzman [3], Ren and Wang [31], and Wang and Wang [41]; moving finite element methods of Millers [30], and Davis and Flaherty [9]; moving mesh PDEs methods of Russell et al. [5,33], Li and Petzold [24], and Ceniceros and Hou [6]; and moving mesh methods based on the harmonic mapping of Dvinsky [14], and Li, Tang and Zhang [10,22,23]. Computational costs of moving mesh methods can be efficiently saved with locally varying time steps [34]. Balanced monitoring of flow phenomena in moving mesh method is recently discussed in [39]. We also refer the readers to recent review papers [4,38] and references therein.

The paper is organized as follows. Section 2 introduces the governing equations of the (ideal) relativistic hydrodynamics and gives the recovering procedure of the primitive variables from the conservative variables, which is a key ingredient in numerical algorithms of the RHDs. Section 3 is devoted to present the basic time evolution algorithm for the two-dimensional (2D) RHD equations. It is a high-resolution finite volume scheme on (general) quadrilateral meshes. Section 4 shows our moving mesh strategies including the mesh iteration redistribution and the conservative interpolation for the conservative variables. The full solution procedure will be outlined in Section 5. Section 6 conducts several numerical experiments to demonstrate the efficiency of the proposed adaptive moving mesh method. They are 2D smooth problem for the accuracy test, quasi 2D relativistic Riemann problems, genuinely 2D relativistic Riemann problems, the relativistic double Mach reflection problem, and the shock-bubble interaction problems. Several concluding remarks are given in Section 7.

2 Governing equations

As the motion of the ideal compressible non-relativistic fluids, the motion of an ideal relativistic fluid is also governed by the continuum equation, the momentum conservation, and the energy conservation. The covariant form of four-dimensional space-time relativistic hydrodynamic (RHD) equations is given as follows [20]

$$\begin{cases} \partial_\alpha(\rho u^\alpha)=0, \\ \partial_\alpha\left(\rho h u^\alpha u^\beta + p g^{\alpha\beta}\right)=0, \end{cases} \tag{2.1}$$

where the Greek indices α and β run from 0 to 3, $\partial_\alpha=\partial_{x^\alpha}$ stands for the covariant derivative, u^α denotes the four-velocity vector, the metric tensor in this paper is restricted to the Minkowski tensor, i.e., $\left(g^{\alpha\beta}\right)_{4\times 4}=\mathrm{diag}\{-1,1,1,1\}$, ρ and p denote the rest-mass density and the kinetic pressure, respectively, and h is the specific enthalpy defined by

$$h=1+e+\frac{p}{\rho}. \tag{2.2}$$

Here e is the specific internal energy. Throughout the paper, units in which the speed of light is equal to one will be used so that

$$x^\alpha=(t,x_1,x_2,x_3)^T \quad \text{and} \quad u^\alpha=\gamma(1,v_1,v_2,v_3)^T,$$

where $\gamma=1/\sqrt{1-v^2}$ is the Lorentz factor and $v^2:=v_1^2+v_2^2+v_3^2$.

An equation of state for the thermodynamical variables is needed to close the system of Eq. (2.1). Assuming that the fluid is a perfect gas so that the equation of state can then be expressed in the form of the algebraic equation

$$p=(\Gamma-1)\rho e, \tag{2.3}$$

where Γ is the adiabatic index, taken as $5/3$ for the mildly relativistic case and $4/3$ for the ultra-relativistic case, i.e., $\rho e \gg 0$, see [45].

In order to solve the RHD equations using the conservative shock-capturing methods, the covariant form (2.1) is casted into a system of conservation laws as follows

$$\frac{\partial \boldsymbol{U}}{\partial t}+\sum_{i=1}^{3}\frac{\partial \boldsymbol{F}_i(\boldsymbol{U})}{\partial x_i}=0, \tag{2.4}$$

where $\boldsymbol{U}$ and $\boldsymbol{F}_i$ are the conservative variable vector and the flux vector in the x_i-direction, respectively, defined by

$$\begin{aligned} &\boldsymbol{U}=\big(D,m_1,m_2,m_3,E\big)^T, && \boldsymbol{F}_1=\big(Dv_1,m_1v_1+p,m_2v_1,m_3v_1,m_1\big)^T,\\ &\boldsymbol{F}_2=\big(Dv_2,m_1v_2,m_2v_2+p,m_3v_2,m_2\big)^T, && \boldsymbol{F}_3=\big(Dv_3,m_1v_3,m_2v_3,m_3v_3+p,m_3\big)^T. \end{aligned}$$

Here $D=\gamma\rho$, $m_i=Dh\gamma v_i$ and $E=Dh\gamma-p$ denote the mass, x_i-momentum and energy densities relative to the laboratory frame, respectively, $i=1,2,3$. Relations between the laboratory quantities (the mass density D, the momentum density m_i, and the energy density E) and the quantities in the local rest frame (the energy density e, the mass density ρ, and the fluid velocity v_i) introduce a strong coupling between the hydrodynamic equations and pose additional numerical difficulties. In practical computations by using the conservative shock-capturing methods, the primitive variable vector $\boldsymbol{V}:=(\rho,v_1,v_2,v_3,p)^T$ has to be calculated from the known conservative one $\boldsymbol{U}$ at each time step. This is very trivial for the Euler equations of the perfect gas in the non-relativistic hydrodynamics, but the relativistic case is not so trivial as the previous. Assuming that the values of $\boldsymbol{U}$ are given, we can solve a non-linear pressure equation such as

$$E+p=D\gamma+\frac{\Gamma}{\Gamma-1}p\gamma^2, \tag{2.5}$$

by any standard root-finding algorithm, e.g., Newton's method, to get the pressure. Note that the Lorentz factor γ in (2.5) has been rewritten into $\big(1-(m_1^2+m_2^2+m_3^2)/(E+p)^2\big)^{-1/2}$. Then γ, ρ, e, h, and v_i can be calculated sequentially.

In the following, we will restrict our attention to two-dimensional RHD equations, that is

$$\frac{\partial \boldsymbol{U}}{\partial t}+\sum_{i=1}^{2}\frac{\partial \boldsymbol{F}_i(\boldsymbol{U})}{\partial x_i}=0, \tag{2.6}$$

where the conservative variable vector $\boldsymbol{U}$ and the flux vector in the x_i-direction $\boldsymbol{F}_i$, $i=1,2$, become

$$\boldsymbol{U}=\big(D,m_1,m_2,E\big)^T,\quad \boldsymbol{F}_1=\big(Dv_1,m_1v_1+p,m_2v_1,m_1\big)^T,\quad \boldsymbol{F}_2=\big(Dv_2,m_1v_2,m_2v_2+p,m_2\big)^T.$$

Generally, numerical computations still need the eigenvalues of the Jacobian matrix $\mathcal{B}_i=\partial\boldsymbol{F}_i/\partial\boldsymbol{U}$ corresponding to the flux vector $\boldsymbol{F}_i$ in (2.6), $i=1,2$, which have been discussed in the literatures, see e.g., [17]. Here, as an example, we only list four eigenvalues of the Jacobian matrix $\mathcal{B}_1$

$$\begin{aligned}
\lambda_1^{(1)} &= \frac{v_1(1-c_s^2)-c_s\sqrt{(1-v^2)\left[1-v_1^2-(v^2-v_1^2)c_s^2\right]}}{1-v^2c_s^2},\\
\lambda_1^{(2)} &= \lambda_1^{(3)} = v_1,\\
\lambda_1^{(4)} &= \frac{v_1(1-c_s^2)+c_s\sqrt{(1-v^2)\left[1-v_1^2-(v^2-v_1^2)c_s^2\right]}}{1-v^2c_s^2},
\end{aligned}$$

where c_s denotes the sound speed, defined by

$$c_s = \sqrt{\Gamma p/\rho h}. \tag{2.7}$$

In the limit $|v_1| \nrightarrow 1$, the 1st and 4th characteristic fields are genuinely nonlinear and others are linearly degenerate. But, in the limit $|v_1| \rightarrow 1$, the 1st and 4th characteristic fields will become linearly degenerate. The eigenvalues of the Jacobian matrix $\mathcal{B}_2$ can be similarly obtained by using symmetry considerations. In the one-dimensional case, the velocity component v_2 disappears and two genuinely nonlinear characteristic fields reduce to $(v_1 \mp c_s)/(1 \mp v_1 c_s)$.

For completeness, Appendix A shows the procedure of deriving the eigenvalues and the eigenvectors of the Jacobian matrix $\mathcal{B}_i = \partial \boldsymbol{F}_i/\partial \boldsymbol{U}$.

3 Basic numerical scheme

This section is devoted to present the basic time evolution algorithm for the two-dimensional RHD equations (2.6). It is a high-resolution finite volume scheme on (general) quadrilateral meshes.

Denote the coordinate (x_1,x_2) by $\boldsymbol{x}$ and given a partition of the physical domain Ω_p as $\{A_{j+\frac{1}{2},k+\frac{1}{2}} | A_{j+\frac{1}{2},k+\frac{1}{2}} \subset \Omega_p, j,k \in \mathbb{Z}\}$ and a partition of the time interval $[0,T]$, $\{t_{n+1} = t_n + \Delta t_n | \Delta t_n > 0,\ n \in \mathbb{N}\}$, where $A_{j+\frac{1}{2},k+\frac{1}{2}}$ is a quadrangle with four vertexes $\boldsymbol{x}_{j,k}, \boldsymbol{x}_{j+1,k}, \boldsymbol{x}_{j+1,k+1}$ and $\boldsymbol{x}_{j,k+1}$, see its schematic interpretation in Fig. 1 and the time step size Δt_n will be suitably chosen following the stability requirement. We will use $\mathcal{L}^{(\ell)}$ to denote the ℓth edge of $A_{j+\frac{1}{2},k+\frac{1}{2}}$ and $\boldsymbol{n}^{(\ell)} = (n_1^{(\ell)}, n_2^{(\ell)})$ be the outward unit normal vector on the ℓth edge of the control volume $A_{j+\frac{1}{2},k+\frac{1}{2}}$, $\ell = 1,2,3,4$. For example, the edge $\mathcal{L}^{(1)}$ is the line segment with two end points $\boldsymbol{x}_{j,k}$ and $\boldsymbol{x}_{j+1,k}$, and then other edges are ordered sequentially in an anticlockwise direction.

Integrating the Eq. (2.6) over the control volume $A_{j+\frac{1}{2},k+\frac{1}{2}}$ and using the divergence theorem gives

$$|A_{j+\frac{1}{2},k+\frac{1}{2}}| \frac{\mathrm{d}}{\mathrm{d}t} \overline{\boldsymbol{U}}_{j+\frac{1}{2},k+\frac{1}{2}}(t) + \sum_{\ell=1}^{4} \int_{\mathcal{L}^{(\ell)}} \boldsymbol{F}_{n^{(\ell)}}(\boldsymbol{U}) \mathrm{d}s = 0, \tag{3.1}$$

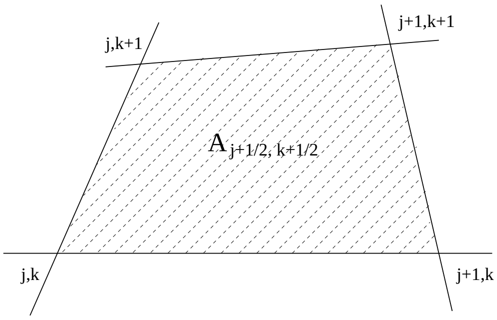

Figure 1: A schematic interpretation of $A_{j+\frac{1}{2},k+\frac{1}{2}}$.

where $|A_{j+\frac{1}{2},k+\frac{1}{2}}|$ denotes the area of the control volume $A_{j+\frac{1}{2},k+\frac{1}{2}}$, $\boldsymbol{F}_{n^{(\ell)}}=\boldsymbol{F}_1 n_1^{(\ell)}+\boldsymbol{F}_2 n_2^{(\ell)}$, and $\overline{\boldsymbol{U}}_{j+\frac{1}{2},k+\frac{1}{2}}(t)$ is the cell average of the conservative variable vector $\boldsymbol{U}(\boldsymbol{x},t)$ over the cell $A_{j+\frac{1}{2},k+\frac{1}{2}}$, i.e.,

$$\overline{\boldsymbol{U}}_{j+\frac{1}{2},k+\frac{1}{2}}(t)=\frac{1}{|A_{j+\frac{1}{2},k+\frac{1}{2}}|}\int_{A_{j+\frac{1}{2},k+\frac{1}{2}}}\boldsymbol{U}(\boldsymbol{x},t)\mathrm{d}\boldsymbol{x}. \tag{3.2}$$

Eq. (3.1) is the base of the finite volume scheme approximating (2.6) and may be considered as a governing equation for the cell averages $\{\overline{\boldsymbol{U}}_{j+\frac{1}{2},k+\frac{1}{2}}(t)\}$. To derive a finite volume scheme of (2.6), the solution $\boldsymbol{U}$ or the flux $\boldsymbol{F}_{n^{(\ell)}}(\boldsymbol{U})$ appeared in the second term at the left hand side of (3.1) should be approximated appropriately. Usually, a two-point numerical flux $\widehat{\boldsymbol{F}}_{n^{(\ell)}}(\boldsymbol{U}_L,\boldsymbol{U}_R)$ is used to approximate the flux $\boldsymbol{F}_{n^{(\ell)}}(\boldsymbol{U})$ in (3.1) so that we have a semi-discrete numerical scheme for (3.1)

$$\frac{\mathrm{d}}{\mathrm{d}t}\overline{\boldsymbol{U}}_{j+\frac{1}{2},k+\frac{1}{2}}(t)=-\frac{1}{|A_{j+\frac{1}{2},k+\frac{1}{2}}|}\sum_{\ell=1}^{4}\int_{\mathcal{L}^{(\ell)}}\widehat{\boldsymbol{F}}_{n^{(\ell)}}(\boldsymbol{U}_L^{(\ell)},\boldsymbol{U}_R^{(\ell)})\,\mathrm{d}s, \tag{3.3}$$

where $\boldsymbol{U}_L^{(\ell)}$ and $\boldsymbol{U}_R^{(\ell)}$ are defined by

$$\boldsymbol{U}_L^{(\ell)}:=\lim_{\boldsymbol{x}\to\boldsymbol{x}^{(\ell)}-}\boldsymbol{U}_h(\boldsymbol{x},t),\quad \boldsymbol{U}_R^{(\ell)}:=\lim_{\boldsymbol{x}\to\boldsymbol{x}^{(\ell)}+}\boldsymbol{U}_h(\boldsymbol{x},t),\quad \boldsymbol{x}^{(\ell)}\in\mathcal{L}^{(\ell)}.$$

Here $\boldsymbol{U}_h(\boldsymbol{x},t)$ is an approximation of the exact solution $\boldsymbol{U}$, e.g., a piecewise polynomial obtained or reconstructed by using the cell averages $\{\overline{\boldsymbol{U}}_{j+\frac{1}{2},k+\frac{1}{2}}(t)\}$. The numerical flux $\widehat{\boldsymbol{F}}_{n^{(\ell)}}(\boldsymbol{U}_L^{(\ell)},\boldsymbol{U}_R^{(\ell)})$ may be formulated by using either approximate or exact Riemann solvers. For the sake of simplicity, a local Lax-Friedrichs flux

$$\widehat{\boldsymbol{F}}_{n^{(\ell)}}(\boldsymbol{U}_L^{(\ell)},\boldsymbol{U}_R^{(\ell)})=\frac{1}{2}\big(\boldsymbol{F}_{n^{(\ell)}}(\boldsymbol{U}_L^{(\ell)})+\boldsymbol{F}_{n^{(\ell)}}(\boldsymbol{U}_R^{(\ell)})-\boldsymbol{Q}^{(\ell)}(\boldsymbol{U}_R^{(\ell)}-\boldsymbol{U}_L^{(\ell)})\big) \tag{3.4}$$

is employed in the current work, where $\boldsymbol{Q}^{(\ell)}=\boldsymbol{Q}^{(\ell)}(\boldsymbol{U}_L^{(\ell)},\boldsymbol{U}_R^{(\ell)})$ denotes the numerical viscosity matrix, which is assumed to be a diagonal matrix and each of its diagonal elements is not less than $\max_{\boldsymbol{U}}\max_m\{|\lambda^{(m)}_{n^{(\ell)}}|\}$ in order to preserve the numerical stability, and $\lambda^{(m)}_{n^{(\ell)}}$ denotes the mth eigenvalue of the Jacobian matrix $\partial \boldsymbol{F}_{n^{(\ell)}}(\boldsymbol{U})/\partial\boldsymbol{U}$, $m=1,2,3,4$.

Our numerical scheme is restricted to second-order accurate shock-capturing schemes based on the initial reconstruction technique. In this case, the definite integral in (3.3) can be approximately calculated by using the midpoint rule and the approximate solution $\boldsymbol{U}_h(\boldsymbol{x},t)$ is defined as a piecewise linear function of $\boldsymbol{x}$ such as

$$\boldsymbol{U}_h(\boldsymbol{x},t)=\overline{\boldsymbol{U}}_{j+\frac{1}{2},k+\frac{1}{2}}(t)+\overline{\boldsymbol{S}}_{j+\frac{1}{2},k+\frac{1}{2}}(t)\big(\boldsymbol{x}-\boldsymbol{x}_{j+\frac{1}{2},k+\frac{1}{2}}\big),\quad \boldsymbol{x}\in A_{j+\frac{1}{2},k+\frac{1}{2}},\tag{3.5}$$

where $\overline{\boldsymbol{S}}_{j+\frac{1}{2},k+\frac{1}{2}}(t)$ is an approximation of the "gradient" $\nabla\boldsymbol{U}(\boldsymbol{x}_{j+\frac{1}{2},k+\frac{1}{2}},t)$ obtained by using any slope limiter and the cell averages $\{\overline{\boldsymbol{U}}_{j+\frac{1}{2},k+\frac{1}{2}}\}$. After those, the scheme (3.3) is replaced with

$$\frac{\mathrm{d}}{\mathrm{d}t}\overline{\boldsymbol{U}}_{j+\frac{1}{2},k+\frac{1}{2}}(t)=-\frac{1}{\big|A_{j+\frac{1}{2},k+\frac{1}{2}}\big|}\sum_{\ell=1}^{4}\widehat{\boldsymbol{F}}_{n^{(\ell)}}\big(\boldsymbol{U}_L^{(\ell)},\boldsymbol{U}_R^{(\ell)}\big)|\mathcal{L}^{(\ell)}|,\tag{3.6}$$

where $|\mathcal{L}^{(\ell)}|$ is the length of the line segment or the edge $\mathcal{L}^{(\ell)}$, $\boldsymbol{U}_L^{(\ell)}$ and $\boldsymbol{U}_R^{(\ell)}$ for $\ell=1$ and 2 are

$$\begin{aligned}
\boldsymbol{U}_L^{(1)}&=\overline{\boldsymbol{U}}_{j+\frac{1}{2},k+\frac{1}{2}}(t)+\overline{\boldsymbol{S}}_{j+\frac{1}{2},k+\frac{1}{2}}(t)\big(\boldsymbol{x}_{j+\frac{1}{2},k}-\boldsymbol{x}_{j+\frac{1}{2},k+\frac{1}{2}}\big),\\
\boldsymbol{U}_R^{(1)}&=\overline{\boldsymbol{U}}_{j+\frac{1}{2},k-\frac{1}{2}}(t)+\overline{\boldsymbol{S}}_{j+\frac{1}{2},k-\frac{1}{2}}(t)\big(\boldsymbol{x}_{j+\frac{1}{2},k}-\boldsymbol{x}_{j+\frac{1}{2},k-\frac{1}{2}}\big),\\
\boldsymbol{U}_L^{(2)}&=\overline{\boldsymbol{U}}_{j+\frac{1}{2},k+\frac{1}{2}}(t)+\overline{\boldsymbol{S}}_{j+\frac{1}{2},k+\frac{1}{2}}(t)\big(\boldsymbol{x}_{j+1,k+\frac{1}{2}}-\boldsymbol{x}_{j+\frac{1}{2},k+\frac{1}{2}}\big),\\
\boldsymbol{U}_R^{(2)}&=\overline{\boldsymbol{U}}_{j+\frac{3}{2},k+\frac{1}{2}}(t)+\overline{\boldsymbol{S}}_{j+\frac{3}{2},k+\frac{1}{2}}(t)\big(\boldsymbol{x}_{j+1,k+\frac{1}{2}}-\boldsymbol{x}_{j+\frac{3}{2},k+\frac{1}{2}}\big),
\end{aligned}$$

and $\boldsymbol{U}_L^{(\ell)}$ and $\boldsymbol{U}_R^{(\ell)}$ for $\ell=3$ and 4 can be similarly given.

Remark 3.1. In practical computations, the reconstruction of the primitive variables in the logical domain Ω_l will replace the reconstruction of the conservative variables in (3.5) so that the values of the primitive variables $\boldsymbol{V}_L^{(\ell)}$ and $\boldsymbol{V}_R^{(\ell)}$ are calculated instead of $\boldsymbol{U}_L^{(\ell)}$ and $\boldsymbol{U}_R^{(\ell)}$ for the numerical fluxes or the time evolution of the cell averages. There are two advantages of reconstructing the primitive variables instead of the conservative variables. The first is that calculating the primitive variables from the known cell averages of the conservative vector is reduced to once per cell in (3.3). The second is that it is easy to avoid the occurrence of the nonphysical values of the physical variables e.g., the negative pressure and the velocity larger than the speed of light.

Because both $\boldsymbol{F}_{n^{(\ell)}}$ and $\boldsymbol{U}$ are functions of the primitive variables $\boldsymbol{V}$, the numerical flux (3.4) for (3.6) can be rewritten as follows

$$\begin{aligned}
\widehat{\boldsymbol{F}}_{n^{(\ell)}}\big(\boldsymbol{V}_L^{(\ell)},\boldsymbol{V}_R^{(\ell)}\big)=&\frac{1}{2}\big(\boldsymbol{F}_{n^{(\ell)}}\big(\boldsymbol{U}(\boldsymbol{V}_L^{(\ell)})\big)+\boldsymbol{F}_{n^{(\ell)}}\big(\boldsymbol{U}(\boldsymbol{V}_R^{(\ell)})\big)\big)\\
&-\frac{1}{2}\boldsymbol{Q}^{(\ell)}\big(\boldsymbol{U}(\boldsymbol{V}_L^{(\ell)}),\boldsymbol{U}(\boldsymbol{V}_R^{(\ell)})\big)\big(\boldsymbol{U}(\boldsymbol{V}_R^{(\ell)})-\boldsymbol{U}(\boldsymbol{V}_L^{(\ell)})\big),
\end{aligned}$$

with (for example)

$$\boldsymbol{V}_L^{(2)}=\overline{\boldsymbol{V}}_{j+\frac{1}{2},k+\frac{1}{2}}+\frac{1}{2}\delta\boldsymbol{V}^{(2)}_{j+\frac{1}{2},k+\frac{1}{2}},\qquad \boldsymbol{V}_R^{(2)}=\overline{\boldsymbol{V}}_{j+\frac{3}{2},k+\frac{1}{2}}-\frac{1}{2}\delta\boldsymbol{V}^{(2)}_{j+\frac{3}{2},k+\frac{1}{2}}, \tag{3.7}$$

where the primitive variable vector $\overline{\boldsymbol{V}}_{j+\frac{1}{2},k+\frac{1}{2}}$ is calculated from the known cell average $\overline{\boldsymbol{U}}_{j+\frac{1}{2},k+\frac{1}{2}}$. The "approximated slope" $\delta\boldsymbol{V}$ is computed by using the values $\{\overline{\boldsymbol{V}}_{j+\frac{1}{2},k+\frac{1}{2}}\}$ and any slope limiter, such as the minmod limiter, the van Leer limiter, or the monotonized central difference limiter, see [21].

Remark 3.2. Other numerical fluxes can also be used, e.g., the HLL (Harten-Lax-van Leer) flux

$$\hat{\boldsymbol{F}}^{HLL}_{n^\ell}=\begin{cases}\boldsymbol{F}_L, & 0\le\alpha_-\le\alpha_+,\\ \dfrac{\alpha_+\boldsymbol{F}_L-\alpha_-\boldsymbol{F}_R+\alpha_+\alpha_-(\boldsymbol{U}_R^\ell-\boldsymbol{U}_L^\ell)}{\alpha_+-\alpha_-}, & \alpha_-<0<\alpha_+,\\ \boldsymbol{F}_R, & \alpha_-\le\alpha_+\le 0,\end{cases} \tag{3.8}$$

where $\boldsymbol{F}_L=\boldsymbol{F}_{n^\ell}(\boldsymbol{U}_L^\ell)$ and $\boldsymbol{F}_R=\boldsymbol{F}_{n^\ell}(\boldsymbol{U}_R^\ell)$, and the coefficients $\alpha_\pm$ are taken into account the highest speeds of the head and the tail of the Riemann fan, which can be estimated from the minimum eigenvalue $\lambda^{(1)}_{n^{(\ell)}}$ and the maximum eigenvalue $\lambda^{(4)}_{n^{(\ell)}}$, that is

$$\alpha_-=\min\left\{\lambda^{(1)}_{n^{(\ell)}}(\boldsymbol{V}_L^\ell),\lambda^{(1)}_{n^{(\ell)}}(\boldsymbol{V}_R^\ell)\right\},\qquad \alpha_+=\max\left\{\lambda^{(4)}_{n^{(\ell)}}(\boldsymbol{V}_L^\ell),\lambda^{(4)}_{n^{(\ell)}}(\boldsymbol{V}_R^\ell)\right\}.$$

The numerical resolution of the HLL numerical flux (3.8) is similar to the above local Lax-Friedrichs flux. They are more dissipative than the numerical flux based on the exact Riemann solver, but the latter is more complicated and time-consuming.

Remark 3.3. To approximate the system of ordinary differential equations (3.6) in time, we use the second-order accurate Runge-Kutta time discretizations. More precisely, if we rewrite the system (3.6) in a compact form

$$\frac{d}{dt}\boldsymbol{U}=L(t,\boldsymbol{U})=L\big(t,\boldsymbol{U}(\boldsymbol{V})\big)=:\widetilde{L}(t,\boldsymbol{V}),$$

then it may be approximated by the following explicit second-order accurate Runge-Kutta method

$$\boldsymbol{U}^*=\boldsymbol{U}^n+\Delta t_n\widetilde{L}(t_n,\boldsymbol{V}^n), \tag{3.9a}$$

$$\boldsymbol{U}^{n+1}=\frac{1}{2}\big(\boldsymbol{U}^n+\boldsymbol{U}^*+\Delta t_n\widetilde{L}(t_n,\boldsymbol{V}^*)\big), \tag{3.9b}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are the conservative and primitive variable vectors respectively.

4 Mesh iteration redistribution

This section presents the adaptive mesh moving strategies including the iteration redistribution of the mesh points and the conservative interpolation of the conservative variables. We refer the readers to [36] for details.

4.1 Redistribution of the mesh point

Let Ω_l and Ω_p be the two-dimensional (2D) logical domain with coordinates $\boldsymbol{\xi}=(\xi_1,\xi_2)$ and the 2D physical domain with coordinates $\boldsymbol{x}=(x_1,x_2)$, respectively. A one-to-one coordinate transformation from Ω_l to Ω_p is denoted by

$$\boldsymbol{x}=\boldsymbol{x}(\boldsymbol{\xi}), \quad \boldsymbol{\xi}\in\Omega_l. \tag{4.1}$$

We limit our attention to the case of that the physical domain Ω_p is convex and the transformation (4.1) is to find the minimizer of the following "mesh-energy" functional defined by

$$\widetilde{E}[\boldsymbol{x}]=\frac{1}{2}\sum_{i=1}^{2}\int_{\Omega_l}(\widetilde{\nabla}x_i)^T G_i\widetilde{\nabla}x_i \mathrm{d}\boldsymbol{\xi}, \tag{4.2}$$

where $\widetilde{\nabla}=(\partial_{\xi_1},\partial_{\xi_2})^T$, and G_1 and G_2 are two given symmetric positive definite matrices called the monitor functions. More terms can be added to the above functional to control other aspects of the mesh such as the orthogonality and the alignment with a given vector field, see e.g., [2,3]. The corresponding Euler- Lagrange equations of (4.2) are

$$\widetilde{\nabla}\cdot(G_i\widetilde{\nabla}x_i)=0, \quad \boldsymbol{\xi}\in\Omega_l, \quad i=1,2. \tag{4.3}$$

Solving the Euler-Lagrange equations or the mesh equations (4.3) on Ω_l will give directly a coordinate transformation $\boldsymbol{x}=\boldsymbol{x}(\boldsymbol{\xi})$ from the logical domain Ω_l to the physical domain Ω_p. Note that Eq. (4.3) is convenient to be numerically solved since the 2D logical domain is chosen as a unit square in most cases.

In general, the monitor functions depend on the solutions or their derivatives of the underlying governing equations and are one of the most important elements in the adaptive moving mesh methods. Appropriate choice of the monitor will generate the meshes with high quality. The simplest choice of the monitor function is

$$G_i=\mathrm{diag}\{\omega,\omega\} \tag{4.4}$$

introduced by Winslow [44], where ω is a positive weight function. It produces isotropic or nondirectional mesh adaptation. A simple directional monitor function considered in [35] is

$$G_i=\mathrm{diag}\{\omega_1,\omega_2\}, \tag{4.5}$$

where ω_1 and ω_2 are two positive weight functions. The monitor function (4.5) will affect mainly the mesh point redistribution in the direction of the logical coordinate lines and reduces to the Winslow monitor function (4.4) if $\omega_1 = \omega_2$.

In our computations, the mesh equations (4.3) are approximated by the central difference scheme on the uniform square mesh and then solved by using the Gauss-Seidel type iteration method

$$\begin{aligned}&(\omega_1)^{[\nu]}_{j+\frac{1}{2},k}\big(\boldsymbol{x}^{[\nu]}_{j+1,k}-\boldsymbol{x}^{[\nu+1]}_{j,k}\big)-(\omega_1)^{[\nu]}_{j-\frac{1}{2},k}\big(\boldsymbol{x}^{[\nu+1]}_{j,k}-\boldsymbol{x}^{[\nu+1]}_{j-1,k}\big)\\&\quad+(\omega_2)^{[\nu]}_{j,k+\frac{1}{2}}\big(\boldsymbol{x}^{[\nu]}_{j,k+1}-\boldsymbol{x}^{[\nu+1]}_{j,k}\big)-(\omega_2)^{[\nu]}_{j,k-\frac{1}{2}}\big(\boldsymbol{x}^{[\nu+1]}_{j,k}-\boldsymbol{x}^{[\nu+1]}_{j,k-1}\big)=0,\end{aligned}\tag{4.6}$$

where

$$\begin{aligned}(\omega_1)^{[\nu]}_{j\pm\frac{1}{2},k}&=\frac{1}{2}\Big(\big(\omega_1(\overline{\boldsymbol{V}}^{[\nu]}_{j\pm\frac{1}{2},k+\frac{1}{2}})\big)+\big(\omega_1(\overline{\boldsymbol{V}}^{[\nu]}_{j\pm\frac{1}{2},k-\frac{1}{2}})\big)\Big),\\(\omega_2)^{[\nu]}_{j,k\pm\frac{1}{2}}&=\frac{1}{2}\Big(\big(\omega_2(\overline{\boldsymbol{V}}^{[\nu]}_{j+\frac{1}{2},k\pm\frac{1}{2}})\big)+\big(\omega_2(\overline{\boldsymbol{V}}^{[\nu]}_{j-\frac{1}{2},k\pm\frac{1}{2}})\big)\Big).\end{aligned}$$

The mesh iteration is continued until $\|\boldsymbol{x}^{[\nu]}-\boldsymbol{x}^{[\nu+1]}\|<\varepsilon$ or $\nu\geq 3$, where ε is an error tolerance.

Different problems may be equipped with different monitor functions. There are several choices of the monitor functions introduced by the existing works for various problems. We refer the readers to [19, 35–37] and references therein. In this work, the weight functions ω in (4.4) and ω_i in (4.5) are respectively chosen as follows

$$\omega(\boldsymbol{V})=\sqrt{1+\sum_{\ell=1}^{3}\alpha_\ell\Big(\min\Big\{1,\frac{|\widetilde{\nabla}\psi_\ell|}{\beta_\ell\max|\widetilde{\nabla}\psi_\ell|+(1-\beta_\ell)\min|\widetilde{\nabla}\psi_\ell|}\Big\}\Big)^2},\tag{4.7}$$

$$\omega_i(\boldsymbol{V})=\sqrt{1+\sum_{\ell=1}^{3}\alpha_{i\ell}\Big(\min\Big\{1,\frac{|\partial\psi_\ell/\partial\xi_i|}{\beta_\ell\max|\widetilde{\nabla}\psi_\ell|+(1-\beta_\ell)\min|\widetilde{\nabla}\psi_\ell|}\Big\}\Big)^2},\tag{4.8}$$

where ψ_ℓ are some physical variables or their functions and α_ℓ, $\alpha_{i\ell}$, and β_ℓ are some non-negative constants, but $\beta_\ell\in[0,1]$, $\ell=1,2,3$, $i=1,2$.

Remark 4.1. In practice, it is common to use some temporal or spatial smoothing on the monitor function to obtain smoother meshes. One of the reasons for using smoothing is to avoid very singular mesh and/or large approximation errors near those regions where the solution has a large gradient. In this work, the following low pass filter is applied to smooth the monitor function

$$\begin{aligned}(\omega_i)_{j+\frac{1}{2},k+\frac{1}{2}}\leftarrow\frac{1}{4}(\omega_i)_{j+\frac{1}{2},k+\frac{1}{2}}&+\frac{1}{8}\big((\omega_i)_{j+\frac{3}{2},k+\frac{1}{2}}+(\omega_i)_{j+\frac{1}{2},k+\frac{3}{2}}+(\omega_i)_{j-\frac{1}{2},k+\frac{1}{2}}+(\omega_i)_{j+\frac{1}{2},k-\frac{1}{2}}\big)\\&+\frac{1}{16}\big((\omega_i)_{j+\frac{3}{2},k+\frac{3}{2}}+(\omega_i)_{j+\frac{3}{2},k-\frac{1}{2}}+(\omega_i)_{j-\frac{1}{2},k+\frac{3}{2}}+(\omega_i)_{j-\frac{1}{2},k-\frac{1}{2}}\big),\end{aligned}$$

$2\sim 3$ times at each iteration of (4.6).

Remark 4.2. In many flow situations, the discontinuities may initially exist in boundaries or move to boundaries at a later time. As a consequence, the grid points on the boundaries of the physical domain Ω_p should be redistributed in order to improve the quality of the solution near boundaries of Ω_p. This work uses the redistribution strategy for boundary points given in [36].

4.2 Interpolation of the conservative variables

After each iteration of the mesh redistribution by (4.6), the solution information has to be passed from the "old" mesh $\{\boldsymbol{x}_{j,k}^{[\nu]}\}$ to the "new" mesh $\{\boldsymbol{x}_{j,k}^{[\nu+1]}\}$. Use $A_{j+\frac{1}{2},k+\frac{1}{2}}^{[\nu+1]}$ and $A_{j+\frac{1}{2},k+\frac{1}{2}}^{[\nu]}$ to denote the quadrangles (finite control volumes) with four vertices $\{\boldsymbol{x}_{j+p,k+q}^{[\nu+1]}, p,q=0,1\}$ and $\{\boldsymbol{x}_{j+p,k+q}^{[\nu]}, p,q=0,1\}$, respectively.

A conservative interpolation scheme is derived by using a perturbation method in [36] to update the cell average values $\overline{\boldsymbol{U}}_{j+\frac{1}{2},k+\frac{1}{2}}$ on the control volume $A_{j+\frac{1}{2},k+\frac{1}{2}}^{[\nu+1]}$. It is of form

$$\begin{aligned}\left|A_{j+\frac{1}{2},k+\frac{1}{2}}^{[\nu+1]}\right|\overline{\boldsymbol{U}}_{j+\frac{1}{2},k+\frac{1}{2}}^{[\nu+1]}=&\left|A_{j+\frac{1}{2},k+\frac{1}{2}}^{[\nu]}\right|\overline{\boldsymbol{U}}_{j+\frac{1}{2},k+\frac{1}{2}}^{[\nu]}-\left(c_{n^{(1)}}\boldsymbol{U}^{[\nu]}\right)_{j+\frac{1}{2},k}-\left(c_{n^{(2)}}\boldsymbol{U}^{[\nu]}\right)_{j+1,k+\frac{1}{2}}\\&-\left(c_{n^{(3)}}\boldsymbol{U}^{[\nu]}\right)_{j+\frac{1}{2},k+1}-\left(c_{n^{(4)}}\boldsymbol{U}^{[\nu]}\right)_{j,k+\frac{1}{2}},\end{aligned}\tag{4.9}$$

where $\left|A_{j+\frac{1}{2},k+\frac{1}{2}}^{[\nu+1]}\right|$ and $\left|A_{j+\frac{1}{2},k+\frac{1}{2}}^{[\nu]}\right|$ denote the areas of the corresponding control volumes, respectively, and the components of the mesh velocity in the direction $\boldsymbol{n}^{(\ell)}$ are approximately defined by

$$\begin{aligned}c_{n^{(1)}}&=\frac{1}{2}\left(c_{j,k}^1+c_{j+1,k}^1\right)\left((x_2)_{j+1,k}^{[\nu]}-(x_2)_{j,k}^{[\nu]}\right)-\frac{1}{2}\left(c_{j,k}^2+c_{j+1,k}^2\right)\left((x_1)_{j+1,k}^{[\nu]}-(x_1)_{j,k}^{[\nu]}\right),\\c_{n^{(2)}}&=\frac{1}{2}\left(c_{\hat{j},k}^1+c_{\hat{j},k+1}^1\right)\left((x_2)_{\hat{j},k+1}^{[\nu]}-(x_2)_{\hat{j},k}^{[\nu]}\right)-\frac{1}{2}\left(c_{\hat{j},k}^2+c_{\hat{j},k+1}^2\right)\left((x_1)_{\hat{j},k+1}^{[\nu]}-(x_1)_{\hat{j},k}^{[\nu]}\right),\\c_{n^{(3)}}&=\frac{1}{2}\left(c_{j+1,\hat{k}}^1+c_{j,\hat{k}}^1\right)\left((x_2)_{j,\hat{k}}^{[\nu]}-(x_2)_{j+1,\hat{k}}^{[\nu]}\right)-\frac{1}{2}\left(c_{j+1,\hat{k}}^2+c_{j,\hat{k}}^2\right)\left((x_1)_{j,\hat{k}}^{[\nu]}-(x_1)_{j+1,\hat{k}}^{[\nu]}\right),\\c_{n^{(4)}}&=\frac{1}{2}\left(c_{j,k+1}^1+c_{j,k}^1\right)\left((x_2)_{j,k}^{[\nu]}-(x_2)_{j,k+1}^{[\nu]}\right)-\frac{1}{2}\left(c_{j,k+1}^2+c_{j,k}^2\right)\left((x_1)_{j,k}^{[\nu]}-(x_1)_{j,k+1}^{[\nu]}\right),\end{aligned}$$

where $c_{n^{(\ell)}}=\boldsymbol{c}^T\boldsymbol{n}^{(\ell)}$, $\boldsymbol{c}=(c^1,c^2)=\left(x_1^{[\nu]}-x_1^{[\nu+1]},x_2^{[\nu]}-x_2^{[\nu+1]}\right)$, and $\boldsymbol{n}^{(\ell)}$ is the normal outward vector of the ℓth edge of the control volume $A_{j+\frac{1}{2},k+\frac{1}{2}}^{[\nu]}$, $\hat{j}:=j+1$ and $\hat{k}:=k+1$.

In Eq. (4.9), the fluxes $(c_{n^{(\ell)}}\boldsymbol{U}^{[\nu]})_{j+r,k+\frac{1}{2}}$ and $(c_{n^{(\ell)}}\boldsymbol{U}^{[\nu]})_{j+\frac{1}{2},k+s}$, $r,s=0$ or 1, denote the values of $c_{n^{(\ell)}}\boldsymbol{U}^{[\nu]}$ on the corresponding edge of the control volume $A_{j+\frac{1}{2},k+\frac{1}{2}}^{[\nu]}$, where $\ell=1,2,3,4$. They will be approximated by using an upwind scheme. For example, the term $(c_{n^{(2)}}\boldsymbol{U}^{[\nu]})_{j+1,k+\frac{1}{2}}$ may be approximated by

$$\left(c_{n^{(2)}}\boldsymbol{U}^{[\nu]}\right)_{j+1,k+\frac{1}{2}}=\frac{c_{n^{(2)}}}{2}\left(\overline{\boldsymbol{U}}_{j+\frac{3}{2},k+\frac{1}{2}}^{[\nu]}+\overline{\boldsymbol{U}}_{j+\frac{1}{2},k+\frac{1}{2}}^{[\nu]}\right)-\frac{\left|c_{n^{(2)}}\right|}{2}\left(\overline{\boldsymbol{U}}_{j+\frac{3}{2},k+\frac{1}{2}}^{[\nu]}-\overline{\boldsymbol{U}}_{j+\frac{1}{2},k+\frac{1}{2}}^{[\nu]}\right).\tag{4.10}$$

In order to avoid large numerical dissipation, the initial reconstruction technique described in Section 3 will be used in practical computations so that for example, we have the numerical flux

$$(c_{n^{(2)}}\boldsymbol{U}^{[\nu]})_{j+1,k+\frac{1}{2}}=\frac{c_{n^{(2)}}}{2}(\boldsymbol{U}_R^{(2)}+\boldsymbol{U}_L^{(2)})-\frac{|c_{n^{(2)}}|}{2}(\boldsymbol{U}_R^{(2)}-\boldsymbol{U}_L^{(2)}). \tag{4.11}$$

It can be easily verified that the interpolation (4.9) preserves conservation of the conservative variables $\boldsymbol{U}$ in the discrete sense

$$\sum_{j,k}\left|A^{[\nu+1]}_{j+\frac{1}{2},k+\frac{1}{2}}\right|\overline{\boldsymbol{U}}^{[\nu+1]}_{j+\frac{1}{2},k+\frac{1}{2}}=\sum_{j,k}\left|A^{[\nu]}_{j+\frac{1}{2},k+\frac{1}{2}}\right|\overline{\boldsymbol{U}}^{[\nu]}_{j+\frac{1}{2},k+\frac{1}{2}}. \tag{4.12}$$

Remark 4.3. The conservative interpolation scheme can also be constructed from the viewpoint of the geometry, see [19].

5 Solution procedure

This section summarizes the solution procedure of our adaptive mesh strategy for two-dimensional relativistic hydrodynamic (RHD) equations (2.6). It consists of two "independent" parts: the time evolution of the RHD equations and the (static) mesh iteration redistribution. In the first part, the RHD equations are discretized by using a high resolution finite volume scheme on the fixed meshes without the full characteristic decomposition of the governing equations, see Section 3. The second part is an iterative procedure, see Section 4. In each iteration, the mesh points are first redistributed, and then the cell averages of the conservative variables are remapped onto the new mesh in a conservative way.

The solution procedure can be illustrated by the following flowchart:

Step 1 At the time level $n=0$, generate an initial adaptive mesh $\boldsymbol{x}^0_{j,k}$ according to the initial data $\boldsymbol{U}(\boldsymbol{x},0)$ and compute the cell averages of the conservative variables $\overline{\boldsymbol{U}}^0_{j+\frac{1}{2},k+\frac{1}{2}}$.

Step 2 If $n>0$, set $\boldsymbol{x}^{[0]}_{j,k}:=\boldsymbol{x}^n_{j,k}$, $\overline{\boldsymbol{U}}^{[0]}_{j+\frac{1}{2},k+\frac{1}{2}}:=\overline{\boldsymbol{U}}^n_{j+\frac{1}{2},k+\frac{1}{2}}$.

For $\nu=0,1,2,\cdots,\mu-1$, do the following:

(a) Compute and smooth the monitor function $\omega^{[\nu]}_{j+\frac{1}{2},k+\frac{1}{2}}$ and move the mesh point $\boldsymbol{x}^{[\nu]}_{j,k}$ to a new position $\boldsymbol{x}^{[\nu+1]}_{j,k}$ by (4.6).

(b) Update the conservative variables, i.e., compute $\overline{\boldsymbol{U}}^{[\nu+1]}_{j+\frac{1}{2},k+\frac{1}{2}}$ by (4.9) and (4.11).

(c) Recover the primitive variables $\overline{\boldsymbol{V}}^{[\nu+1]}_{j+\frac{1}{2},k+\frac{1}{2}}$ from the conservative variables $\overline{\boldsymbol{U}}^{[\nu+1]}_{j+\frac{1}{2},k+\frac{1}{2}}$.

Step 3 Set $\overline{\boldsymbol{U}}^n_{j+\frac{1}{2},k+\frac{1}{2}}:=\overline{\boldsymbol{U}}^{[\mu]}_{j+\frac{1}{2},k+\frac{1}{2}}$, $\overline{\boldsymbol{V}}^n_{j+\frac{1}{2},k+\frac{1}{2}}:=\overline{\boldsymbol{V}}^{[\mu]}_{j+\frac{1}{2},k+\frac{1}{2}}$ and $\boldsymbol{x}^{n+1}_{j,k}:=\boldsymbol{x}^{[\mu]}_{j,k}$. Then, solve the RHD equations (2.6) as follows:

(a) Compute the characteristic speed and determine the time step size Δt_n according to the CFL condition.

(b) Solve the RHD equations with the second-order accurate finite volume scheme given in Section 3 to get the conservative variables $\overline{\boldsymbol{U}}^{n+1}_{j+\frac{1}{2},k+\frac{1}{2}}$.

(c) Recover the primitive variables $\overline{\boldsymbol{V}}^{n+1}_{j+\frac{1}{2},k+\frac{1}{2}}$ from the conservative variables $\overline{\boldsymbol{U}}^{n+1}_{j+\frac{1}{2},k+\frac{1}{2}}$.

Step 4 If $t_{n+1} < T$, then go to Step 2. Otherwise output the results and stop.

6 Numerical experiments

This section will test our two-dimensional (2D) adaptive moving mesh method for some relativistic hydrodynamic (RHD) problems: a 2D smooth problem, two quasi 2D RHD Riemann problems, three 2D RHD Riemann problems, a 2D RHD double Mach reflection problem, and two RHD shock-bubble interaction problems. In our computations, the local Lax-Friedrichs flux is adopted for the basic finite volume scheme, the monotonized central difference limiter [40] will be chosen for the initial reconstruction in solving different problems, and the CFL number is taken as 0.24 unless otherwise stated. All the simulations are completed under the Windows environment of a personal computer of Lenovo (Intel(R) Core(TM)i5 CPU 650 3.20GHZ 2.8GB RAM). For convenience, we will use the notation (x,y) to replace the previous (x_1,x_2) in this section.

6.1 Accuracy test

Example 6.1 (2D smooth problem). In order to check the accuracy of the proposed adaptive moving mesh method for RHDs, we first solve a 2D smooth problem describing a RHD wave propagating in the physical domain $\Omega_p = [0,2/\sqrt{3}] \times [0,2]$ at an angle of $\alpha = 30^\circ$ relative to the x-axis. The initial conditions are taken as

$$\begin{aligned}&\rho(x,y,0) = 1 + 0.2\sin\big(2\pi(x\cos\alpha + y\sin\alpha)\big),\\ &v_1(x,y,0) = 0.2, \quad v_2(x,y,0) = 0, \quad p(x,y,0) = 1.\end{aligned}$$

The problem has the exact solution in the following form

$$\begin{aligned}&\rho(x,y,t) = 1 + 0.2\sin\big[2\pi\big((x\cos\alpha + y\sin\alpha) - (v_1\cos\alpha + v_2\sin\alpha)t\big)\big],\\ &v_1(x,y,t) = 0.2, \quad v_2(x,y,t) = 0, \quad p(x,y,t) = 1.\end{aligned}$$

The domain Ω_p is assumed to be divided into $N \times 2N$ and the periodic boundary conditions are used. Table 1 gives the relative numerical errors δ_N and the convergence rates R_N at $t = 2$, where

$$\delta_N = \frac{\sum_{j=1}^{N}\sum_{k=1}^{2N}\big|\rho^N_{j,k} - \rho^{\text{exact}}_{j,k}\big|}{\sum_{j=1}^{N}\sum_{k=1}^{2N}\big|\rho^{\text{exact}}_{j,k}\big|}, \tag{6.1}$$

Table 1: Example 6.1: Numerical errors and convergence rates at $t=2$ on the uniform meshes and adaptive moving meshes.

N	Fixed mesh		Moving mesh with $\alpha_{11}=\alpha_{21}=10$		Moving mesh with $\alpha_{11}=\alpha_{21}=20$	
	δ_N	R_N	δ_N	R_N	δ_N	R_N
32	1.83e-03	–	5.44e-03	–	6.13e-03	–
64	4.93e-04	1.89	1.43e-03	1.92	1.74e-03	1.82
128	1.28e-04	1.95	3.54e-04	2.02	4.59e-04	1.92
256	3.18e-05	2.01	8.04e-05	2.14	1.10e-04	2.06

and

$$R_N=\log_2\left(\frac{\delta_{\frac{N}{2}}}{\delta_N}\right). \tag{6.2}$$

In the computations, the ideal equation of state is used with $\Gamma=5/3$, and the monitor function is chosen as (4.5) and (4.8) with $(\psi_1,\psi_2,\psi_3)=(\rho,p,v_1)$, $\alpha_{12}=\alpha_{22}=\alpha_{13}=\alpha_{23}=0$, $(\beta_1,\beta_2,\beta_3)=(1,0,0)$, and two different values for $(\alpha_{11},\alpha_{21})$. The results show that the 2nd order accuracy may be obtained for each case. However, it is worth noting that the accuracy of an adaptive moving mesh algorithm is generally dependent on choice of the monitor function. Fig. 2 displays the adaptive meshes of 32×64 cells at three different times $t=1$, 1.5 and 2, which are obtained by the adaptive moving mesh method with $\alpha_{11}=\alpha_{21}=10$.

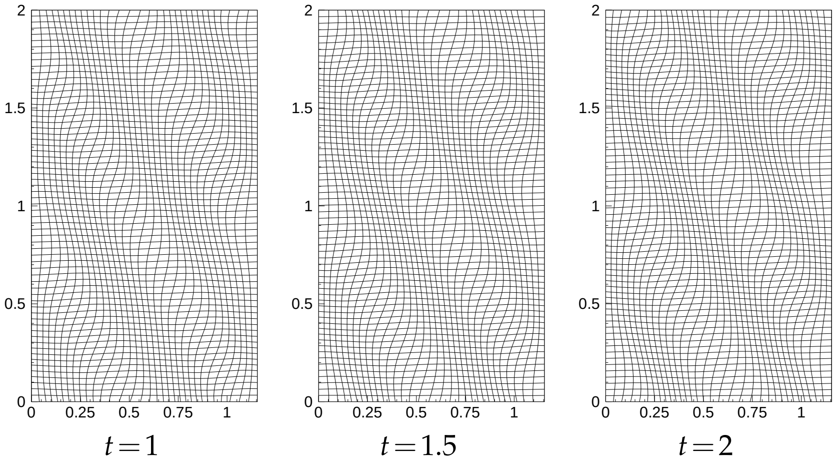

Figure 2: Example 6.1: The adaptive meshes of 32×64 cells at three different times $t=1$, 1.5 and 2.

6.2 Quasi 2D Riemann problems

Next, two quasi 2D RHD Riemann problems are solved by our 2D moving mesh method. Those problems have been considered to validate the 1D shock-capturing schemes for the relativistic hydrodynamics in [25, 28]. In order to verify the ability of the present adaptive moving mesh method, its solutions will be compared to the analytical solutions and the numerical solutions obtained by using the corresponding method on the fixed uniform meshes. The domain Ω_p is taken as $[0,1]\times[0,0.1]$, which is covered by non-uniform 150×10 cells for the adaptive moving mesh method and uniform 400×10 cells respectively. The initial discontinuity of the quasi 2D RHD Riemann problems are located at $x=0.5$, and corresponding exact solutions can be obtained by an iteration procedure, see e.g., [26].

Example 6.2 (Quasi 2D Riemann problem I)**.** The initial data of the first quasi 2D Riemann problem are

$$(\rho,v_1,v_2,p)=\begin{cases}(1,0.9,0,1), & x<0.5,\\ (1,0,0,10), & x>0.5.\end{cases} \tag{6.3}$$

In the computations, the ideal equation of state with $\Gamma=4/3$ is used, and the monitor function is chosen as (4.5) and (4.8) with $(\psi_1,\psi_2,\psi_3)=(\rho,p,v_1)$, $\alpha_{11}=\alpha_{21}=50$, $\alpha_{12}=\alpha_{22}=\alpha_{13}=\alpha_{23}=20$ and $(\beta_1,\beta_2,\beta_3)=(0.1,0.1,0.1)$.

Fig. 3 shows the numerical results at $t=0.4$ by using the adaptive moving mesh method (the left column) and the uniform mesh method (the right column), respectively. The symbol "$\circ$" denotes the numerical solution and the solid line is the exact solution, which consist of a left-moving shock wave, a contact discontinuity, and a right-moving shock wave. Spurious numerical oscillations are observed behind the left-moving shock wave in the solutions of the uniform mesh method, while the adaptive moving mesh method presents the non-oscillatory solution and resolve the discontinuities well.

Example 6.3 (Quasi 2D Riemann problem II)**.** The initial data of the second quasi 2D Riemann problem are

$$(\rho,v_1,v_2,p)=\begin{cases}\big(10,0,0,40/3\big), & x<0.5,\\ (1,0,0,0), & x>0.5.\end{cases} \tag{6.4}$$

The equation of state is the same as the above one except for $\Gamma=5/3$. In this case, the initial discontinuity will be evolved into a rarefaction wave moving to the left and a shock wave moving to the right, and a contact discontinuity in the middle.

Fig. 4 gives the numerical solutions obtained by using the adaptive moving mesh method (the left column) and the uniform mesh method (the right column) respectively, which are compared to the exact solutions. The results show that the adaptive moving mesh method has a better resolution for discontinuities even with lesser number of cells. In the computations, the monitor function is chosen as (4.5) and (4.8) with $(\psi_1,\psi_2,\psi_3)=(\rho,p,v_1)$, $\alpha_{11}=\alpha_{21}=100$, $\alpha_{12}=\alpha_{22}=0$, $\alpha_{13}=\alpha_{23}=0$ and $(\beta_1,\beta_2,\beta_3)=(0.4,1,1)$.

The main differences between the solution of relativistic shock tubes and their non-relativistic counterparts are due to the nonlinear addition of velocities and to the Lorentz

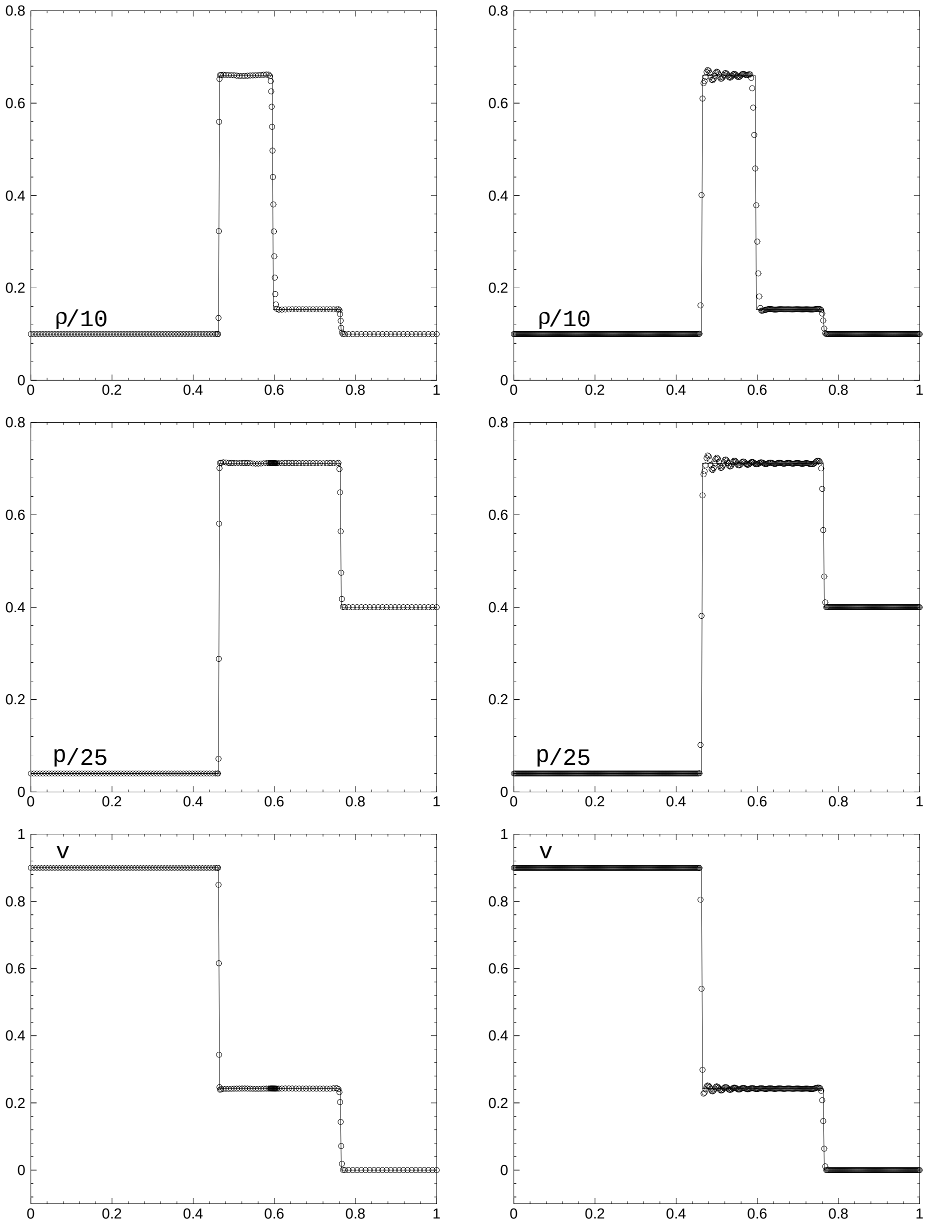

Figure 3: Example 6.2: One-tenth of the density $\rho/10$, one-25th of the pressure $p/25$, and the velocity component in x-direction at $t=0.4$. The solid lines represent the exact solutions. Left: the adaptive moving mesh method with 150×10 cells; right: the uniform mesh method with 400×10 cells.

contraction. The former yields a curved profile for the rarefaction fan, as opposed to a linear one in the non-relativistic case, while the latter narrows the shock plateau. These effects, especially the latter, become particularly noticeable in the ultra-relativistic regime (i.e., $\gamma\gg 1$).

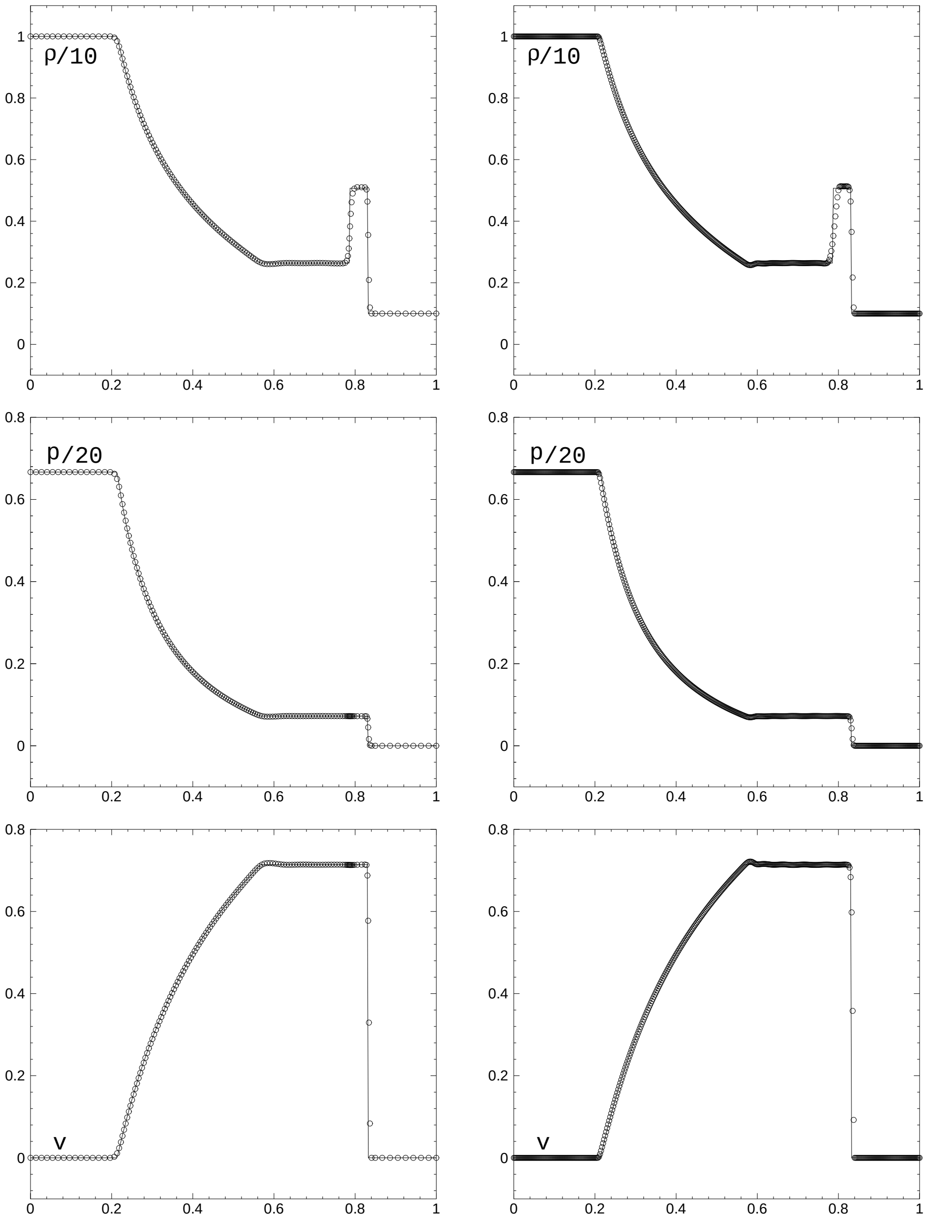

Figure 4: Example 6.3: One-tenth of the density $\rho/10$, one-twentieth of the pressure $p/20$, and the velocity component in the x-direction at $t=0.4$. The solid lines represent the exact solutions. Left: the adaptive moving mesh method with 150×10 cells; right: the uniform mesh method with 400×10 cells.

6.3 Two-dimensional problems

This section verifies the capability of the current adaptive moving mesh scheme in capturing complex 2D wave configurations by solving several genuinely 2D problems of the RHD equations (2.6), including three relativistic 2D Riemann problems, a relativistic double Mach reflection problem, and two relativistic shock-bubble interaction problems.

Example 6.4 (2D Riemann problem I)**.** The 2D Riemann problems of the compressible Euler equations in the non-relativistic hydrodynamics have been widely studied and solved in the literatures, see e.g., [18] and references therein, since they can simulate almost all essential 2D physical phenomena such as shock reflections, the spiral formation (rolling up of slip lines), vortex and shock interactions etc.

The initial data of the first 2D Riemann problem for the ideal relativistic fluid with the adiabatic index $\Gamma=5/3$ are

$$(\rho,v_1,v_2,p)=\begin{cases}(0.035145216124503,0,0,0.162931056509027), & x>0,\ y>0,\\ (0.1,0.7,0,1), & x<0,\ y>0,\\ (0.5,0,0,1), & x<0,\ y<0,\\ (0.1,0,0.7,1), & x>0,\ y<0,\end{cases} \tag{6.5}$$

where the left and bottom discontinuities are contact discontinuities and the top and right are two shock waves with the speed of 0.934563275373844.

In the computation, the physical domain Ω_p is taken as $[-1,1]\times[-1,1]$ with four outflow conditions. The monitor function is chosen as the directional monitor function (4.5) and (4.8) with $(\psi_1,\psi_2,\psi_3)=(\log\rho,p,v^2)$, $\alpha_{11}=\alpha_{21}=70$, $\alpha_{12}=\alpha_{22}=30$, $\alpha_{13}=\alpha_{23}=0$, and $(\beta_1,\beta_2,\beta_3)=(0.1,0.3,1)$. Fig. 5 gives the adaptive mesh of 150×150 cells and the schlieren image of the density at $t=0.8$ obtained by using the adaptive moving mesh method. Fig. 6 presents a comparison of the density logarithms $\log\rho$ along the symmetric axis $y=x$ calculated by using the adaptive moving mesh method and the uniform mesh method, where the symbol "$\circ$" and the solid line denote the adaptive solution with a resolution of 150×150 and the computed solution obtained on a 600×600 uniform mesh. Moreover, the interval of the horizontal axis has been scaled as $[-1,1]$. Table 2 shows the recorded CPU times of the adaptive moving mesh method and the uniform mesh method. It can be seen that the adaptive moving mesh is very effective, the adaptive mesh agrees well with the important features of the solution; the interaction of the initial shock waves and contact discontinuities produces the formation of a "mushroom pattern" round the point $(-0.1,-0.1)$.

Table 2: Examples 6.4-6.6: The estimated CPU times (seconds).

Problem	Algorithm	Mesh number	CPU time
Example 6.4	Moving mesh	150×150	510.6
	Fixed mesh	600×600	2057.4
Example 6.5	Moving mesh	150×150	555.7
	Fixed mesh	450×450	1034.1
Example 6.6	Moving mesh	150×150	326.9
	Fixed mesh	400×400	562.1

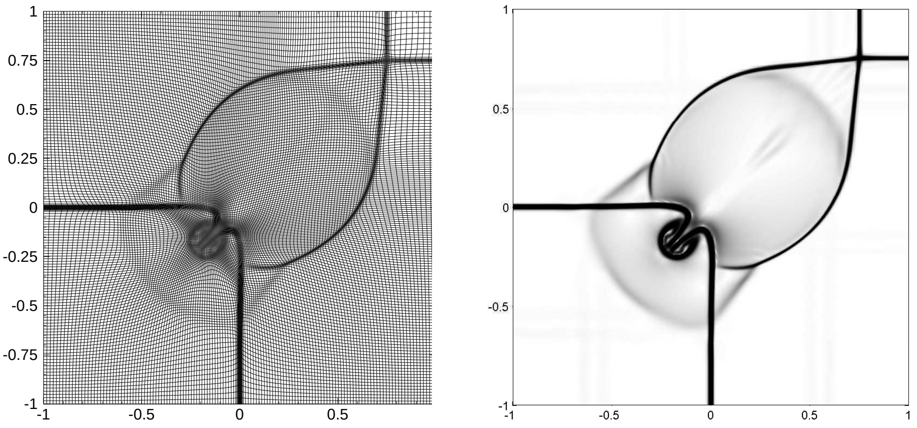

Figure 5: Example 6.4: The adaptive mesh of 150×150 cells and the schlieren image of the density at $t=0.8$.

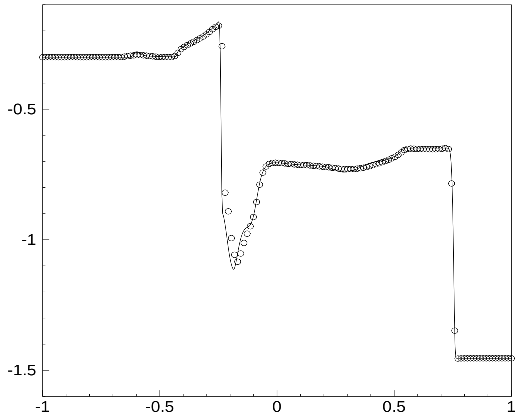

Figure 6: Example 6.4: The density logarithms $\log\rho$ along the symmetric axis $y=x$ at $t=0.8$. The solid line and the symbols "○" denote the solutions obtained on a 600×600 uniform mesh, and a 150×150 moving mesh, respectively.

Example 6.5 (2D Riemann problem II)**.** The second 2D Riemann problem is about the interaction of four vortex sheets (contact discontinuities) for the ideal relativistic fluid with the adiabatic index $\Gamma=5/3$. The initial data are

$$(\rho,v_1,v_2,p)=\begin{cases}(0.5,0.5,-0.5,5), & x>0,\ y>0,\\(1,0.5,0.5,5), & x<0,\ y>0,\\(3,-0.5,0.5,5), & x<0,\ y<0,\\(1.5,-0.5,-0.5,5), & x>0,\ y<0.\end{cases}\tag{6.6}$$

Those four initial vortex sheets with the same sign (the negative sign) will interact each other to form a spiral with the low density in its center $(0,0)$. This is the typical cavitation phenomenon in gas dynamics.

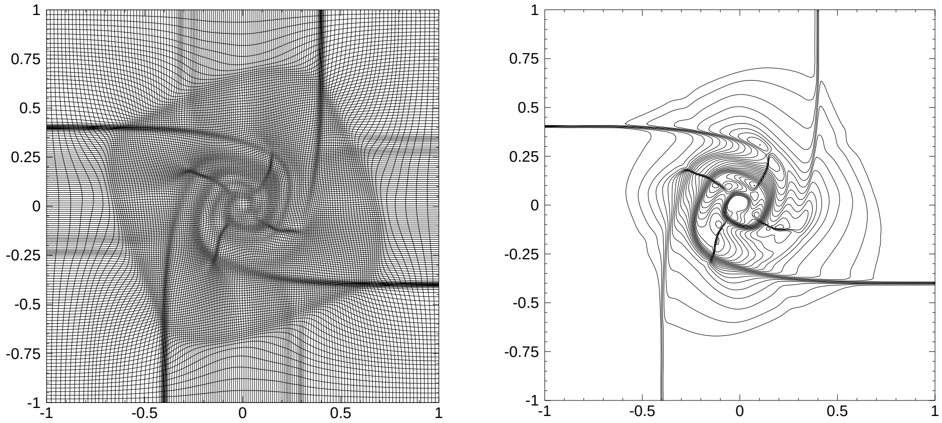

Figure 7: Example 6.5: The adaptive mesh of 150×150 cells (Left) and the contour of the density logarithm (Right) at $t=0.8$ by the adaptive moving mesh method.

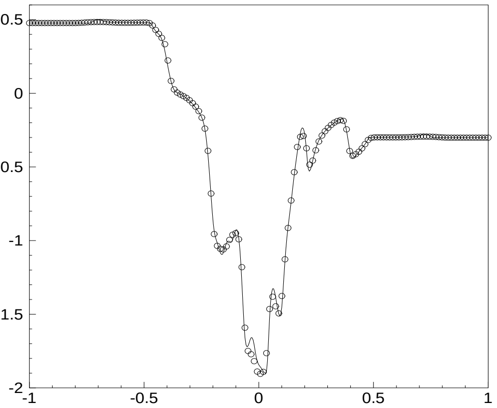

Figure 8: Example 6.5: The density logarithms $\log\rho$ along $y=x$ at $t=0.8$. The solid line and the symbol "$\circ$" denote the numerical solutions obtained on a 450×450 uniform mesh, and a 150×150 moving mesh, respectively.

Fig. 7 displays the adaptive mesh of 150×150 cells and the contour of the density logarithm at $t=0.8$ obtained by using the adaptive moving mesh method. Fig. 8 gives a comparison of the density logarithms $\log\rho$ along the line $y=x$ calculated by using the adaptive moving mesh method and the uniform mesh method, where the symbol "$\circ$" and the solid line denote the adaptive solution with a resolution of 150×150 and the computed solution obtained on a 450×450 uniform mesh, respectively. Their CPU times are given in Table 2. The numerical results exhibit the perfect performance of the scheme in capturing spirals. It can also be seen that the adaptive mesh agrees well with the important features of the solution. Our computations take Ω_p as $[-1,1]\times[-1,1]$, and chooses the monitor function (4.5) and(4.8) with $(\psi_1,\psi_2,\psi_3)=(\log\rho,p,v^2)$, $\alpha_{11}=\alpha_{21}=50$, $\alpha_{12}=\alpha_{22}=50$, $\alpha_{13}=\alpha_{23}=50$ and $(\beta_1,\beta_2,\beta_3)=(0.3,1,0.5)$.

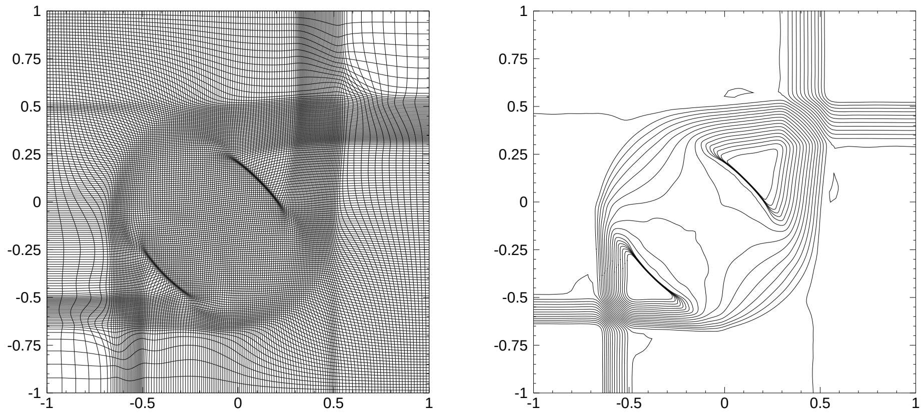

Figure 9: Example 6.6: The adaptive mesh of 150×150 cells (Left) and the contour of the density logarithm (Right) at $t=0.8$ by the adaptive moving mesh method.

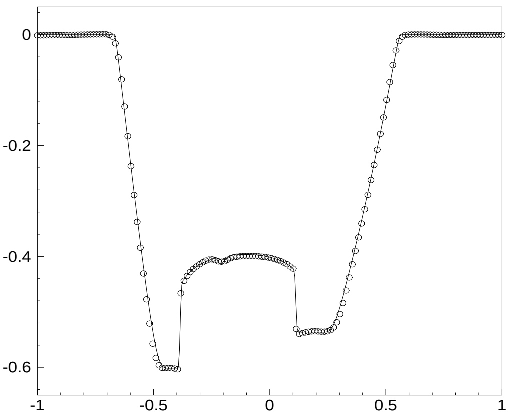

Figure 10: Example 6.6: The density logarithms at $t=0.8$ along $y=x$. The solid line and the symbol "$\circ$" denote the solutions obtained by using the uniform mesh method with 400×400 cells and the adaptive moving mesh method with 150×150 cells, respectively.

Example 6.6 (2D Riemann problem III)**.** The third 2D Riemann problem is about the interaction of pure planar rarefaction waves for the ideal relativistic fluid with the adiabatic index $\Gamma=5/3$. The initial data are chosen as

$$(\rho,v_1,v_2,p)=\begin{cases}(1,0,0,1), & x>0,\ y>0,\\ (0.5771,-0.3529,0,0.4), & x<0,\ y>0,\\ (1,-0.3529,-0.3529,1), & x<0,\ y<0,\\ (0.5771,0,-0.3529,0.4), & x>0,\ y<0.\end{cases} \tag{6.7}$$

Each of the initial discontinuities will evolve into a rarefaction wave, and then four rarefaction waves interact each other and generate two symmetric shock waves in the interaction region after the mutual penetration of the planar rarefaction waves.

Fig. 9 shows the adaptive mesh of 150×150 cells and the contour of the density logarithm at $t=0.8$ by the adaptive moving mesh method. Fig. 10 gives a comparison of the density logarithms $\log\rho$ along the symmetric axis $y=x$ calculated by using the uniform mesh method with 400×400 cells and the adaptive moving mesh method with 150×150 cells, respectively. Their CPU times are given in Table 2. We see that the results obtained by the two methods are in good agreement, and the adaptive moving mesh method is more effective than the uniform mesh method.

In the computation, the physical domain Ω_p is taken as $[-1,1]\times[-1,1]$ with four outflow conditions. The monitor function is chosen as (4.5) and (4.8) with $(\psi_1,\psi_2,\psi_3)=(\log\rho,p,v^2)$, $\alpha_{11}=\alpha_{21}=50$, $\alpha_{12}=\alpha_{22}=100$, $\alpha_{13}=\alpha_{23}=0$ and $(\beta_1,\beta_2,\beta_3)=(0.3,0.3,1)$.

Example 6.7 (Double Mach reflection)**.** The double Mach reflection of a strong shock has often been used to test non-relativistic hydrodynamic codes. Zhang et al. [46] extended such problem to the ideal relativistic fluid with the adiabatic index $\Gamma=1.4$. We use exactly the same setup as in [46], i.e., the same initial and boundary conditions and same domain $\Omega_p=[0,4]\times[0,1]$. Initially a right-moving shock with the speed of v_s is positioned at $x=1/6, y=0$ and makes a 60° angle with the x-axis. More precisely, the initial data for the primitive variables $\boldsymbol{V}=(\rho,v_1,v_2,p)^T$ are set by

$$\boldsymbol{V}(x,y,0)=\left\{\begin{array}{ll}\boldsymbol{V}_L, & y>h(x,0),\\ \boldsymbol{V}_R, & y<h(x,0),\end{array}\right. \tag{6.8}$$

where the left and right states of the shock wave, and the exact position of the shock at the time t are, respectively,

$$\begin{aligned}&\boldsymbol{V}_L=\big(8.564,0.4247\sin60^\circ,-0.4247\cos60^\circ,0.3808\big)^T,\\ &\boldsymbol{V}_R=\big(1.4,0,0,0.0025\big)^T,\\ &h(x,t)=\sqrt{3}\Big(x-\frac{1}{6}\Big)-2v_st, \quad v_s=0.4984.\end{aligned}$$

The boundary conditions are specified as follows: the right boundary is set to the exact pre-shock condition $\boldsymbol{V}_R$; for the top boundary $y=1$, the boundary conditions are set to either the post-shock state $\boldsymbol{V}_L$ or the pre-shock state $\boldsymbol{V}_R$, depending on the exact motion of the shock wave. A reflecting wall is placed at the $x>1/6$ part of the bottom boundary; and the states on the remainder boundaries, i.e., the $x<1/6$ part of the bottom boundary and the left boundary, are set to the exact post-shock state $\boldsymbol{V}_L$.

Fig. 11 shows the adaptive mesh of 200×50 cells at $t=4$ for the whole physical domain $\Omega_p=[0,4]\times[0,1]$. It is observed that lesser mesh points are distributed for $x>3$, which is a desired feature for this test problem. Fig. 12 displays the contours of the density at $t=4$ within the sub-domain $[0,3]\times[0,1]$ computed by using the uniform mesh method with 640×160 cells and the adaptive moving mesh method with 200×50 cells, respectively. The comparison of their CPU times is given in Table 3. It is seen that both methods

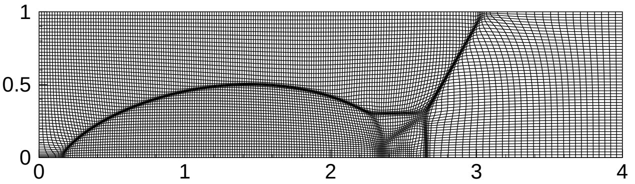

Figure 11: Example 6.7: The adaptive moving mesh of 200×50 cells at $t=4$.

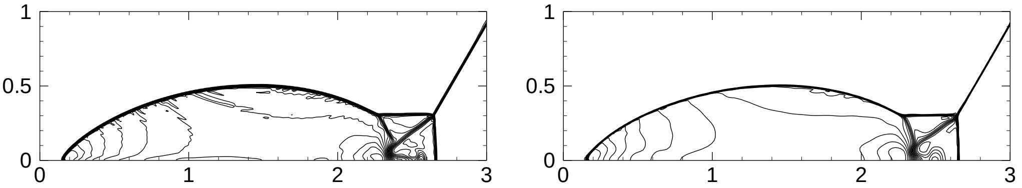

Figure 12: Example 6.7: The contours of the density at $t=4$. The left is obtained on a 640×160 uniform mesh; the right is obtained by the adaptive moving mesh method with 200×50 cells.

Table 3: Example 6.7: The estimated CPU times (seconds).

Algorithm	Mesh number	CPU time
Moving mesh	200×50	519.4
Fixed mesh	640×160	1133.2

produce almost same numerical results, but the adaptive moving mesh method takes much less CPU time than the uniform mesh method.

In the computations, the monitor function is chosen as (4.4) and (4.7) with $(\psi_1,\psi_2,\psi_3)=(\log\rho,p,v^2)$, $(\alpha_1,\alpha_2,\alpha_3)=(100,0,0)$ and $(\beta_1,\beta_2,\beta_3)=(0.15,1,1)$.

Example 6.8 (Shock Bubble Interaction Problem I)**.** The interaction between the shock wave and the bubble in non-relativistic hydrodynamics have been extensively studied by many authors, see e.g., [7]. We extend such problem to the relativistic hydrodynamics. The setup of the problem is as follows. The physical domain Ω_p is taken as $[0,325]\times[-45,45]$ with two reflective boundaries at $y=\pm 45$, and the states on two boundaries in x-direction are set to the left and right shock state, respectively. Initially, a left-moving relativistic straight shock wave is located at $x=265$ with the left and right states

$$(\rho,v_1,v_2,p)=\begin{cases}(1,0,0,0.05), & x<265,\\ (1.865225080631180,-0.196781107378299,0,0.15), & x>265,\end{cases}$$

for the ideal relativistic fluid with the adiabatic index $G=5/3$. There is a cylindrical bubble centered at (215,0) with the radius of 25 in front of the initial shock wave. The initial state of the fluid within the bubble is taken as $(\rho,v_1,v_2,p)=(0.1358,0,0,0.05)$, which is in a mechanical equilibrium with the surrounding fluid and lighter than the ambient fluid.

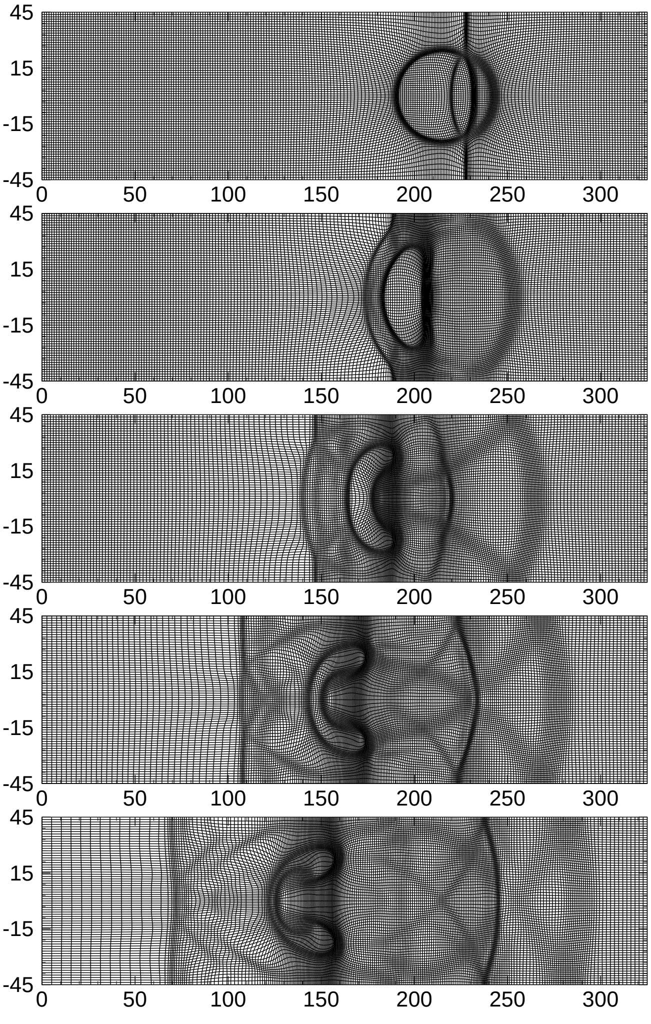

Figure 13: Example 6.8: The adaptive meshes of 260×72 cells at $t=90,180,270,360,450$, respectively.

Figs. 13 and 14 show the adaptive meshes of 260×72 cells and the schlieren images of the density ρ at $t=90,180,270,360$, and 450, respectively, obtained by using the adaptive moving mesh method. Those plots are clearly showing the dynamics of the interaction between the shock wave and the bubble. Our computations choose the monitor functions (4.4) and (4.7) with $(\psi_1,\psi_2,\psi_3)=(\rho,\log\rho,\gamma)$, $(\alpha_1,\alpha_2,\alpha_3)=(20,50,10)$ and $(\beta_1,\beta_2,\beta_3)=(0.05,0.6,0.8)$.

The computational efficiency of the adaptive moving mesh method is compared to the fixed mesh method via the plots of the density along the line $y=0$ given in Fig. 15 and the recorded CPU times presented in Table 4. The solid line and the symbols "$\circ$" denote the solutions obtained on a 650×180 uniform mesh and a 260×72 moving mesh,

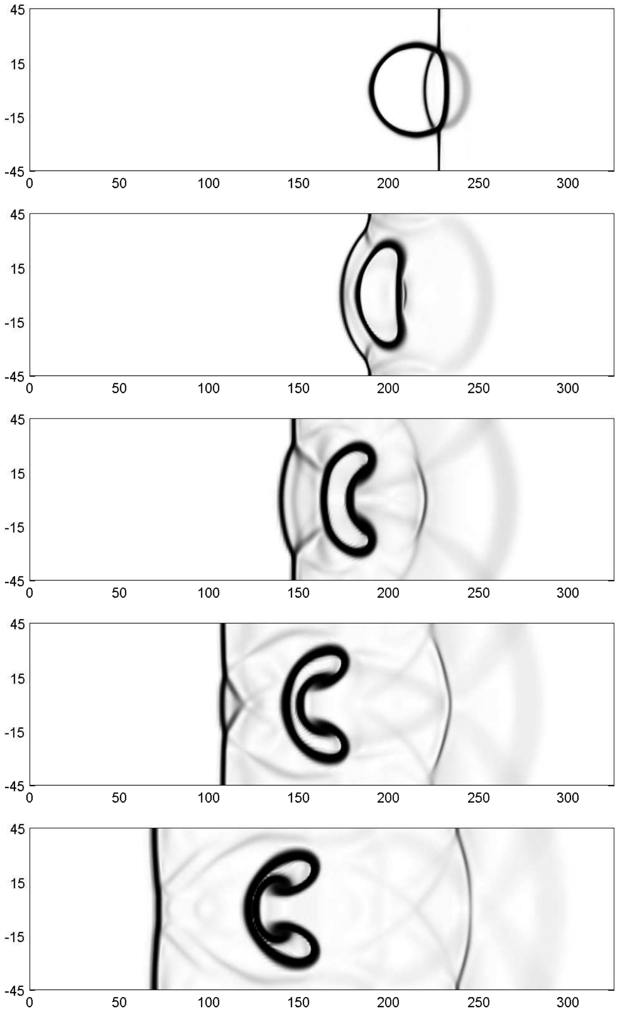

Figure 14: Same as Fig. 13 except for the schlieren images of the density.

respectively. Those results show that the adaptive redistribution of the mesh points can efficiently improve the quality of the computed solution; the discontinuities are captured well and accurately; and at the same time, some small wave structures are also resolved clearly.

Table 4: Example 6.8: The estimated CPU times (seconds).

Algorithm	Mesh number	CPU time
Moving mesh	260×72	1254.9
Fixed mesh	650×180	1829.0

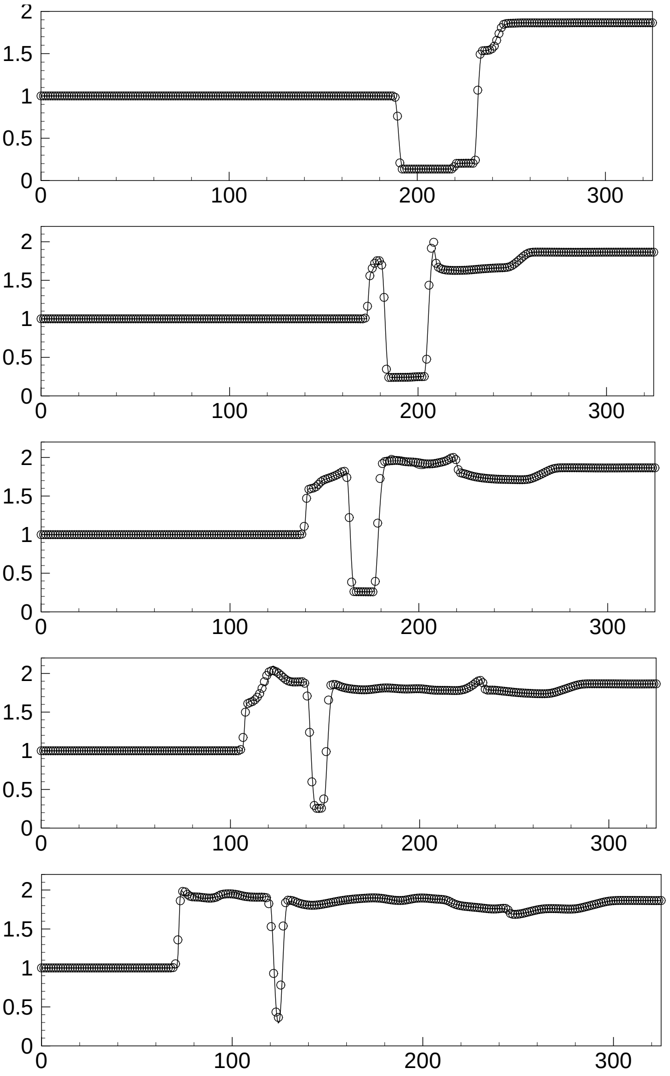

Figure 15: Same as Fig. 13 except for densities ρ along $y=0$. The solid line and the symbols "$\circ$" denote the solutions obtained on a 650×180 uniform mesh and a 260×72 moving mesh, respectively.

Example 6.9 (Shock Bubble Interaction Problem II)**.** This example is similar to the last one, but the present fluid in the bubble is heavier than the ambient fluid. Such difference will yield different flow patterns around the bubble interface after its interaction with the shock, just like in non-relativistic hydrodynamics, see e.g., [7].

The setup of the problem is same as that of the last example, except for that the initial states of the fluid in the bubble $(\rho,v_1,v_2,p)=(3.1538,0,0,0.05)$. Figs. 16 and 17 show the adaptive meshes of 260×72 cells and the schlieren images of the density ρ at t=100,200,300,400, and 500, respectively, obtained by the adaptive moving mesh method.

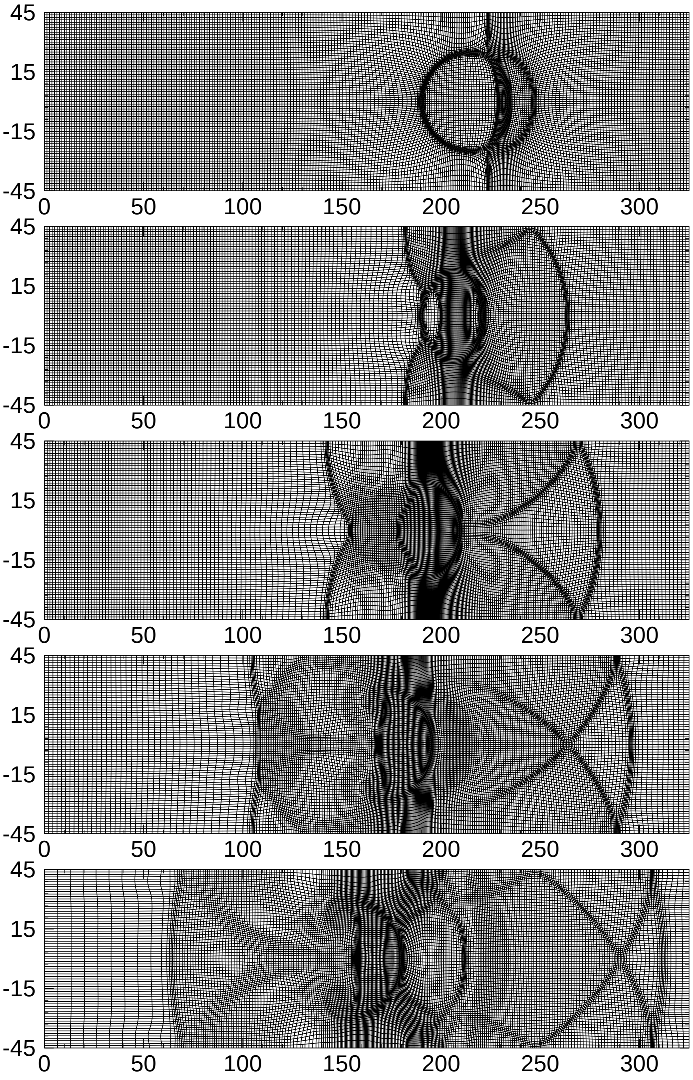

Figure 16: Example 6.9: The adaptive meshes of 260×72 cells at $t=100,200,300,400,500$, respectively.

We see obviously different wave patterns of the interaction between the shock and the bubble, and that the mesh points are well distributed and clustered round the discontinuities so as to effectively improve the quality of the computed solution. The monitor functions (4.4) and (4.7) has been employed to adaptively redistribute the mesh points, where $(\psi_1,\psi_2,\psi_3)=(\rho,\log\rho,\gamma)$, $(\alpha_1,\alpha_2,\alpha_3)=(20,50,10)$, and $(\beta_1,\beta_2,\beta_3)=(0.015,0.6,0.8)$.

A comparison of the adaptive moving mesh method to the uniform mesh method is given via the plots of the density along $y=0$ in Fig. 18 and the recorded CPU times in Table 5. We see from Figs. 15 and 18, and Tables 4 and 5 that the present adaptive moving

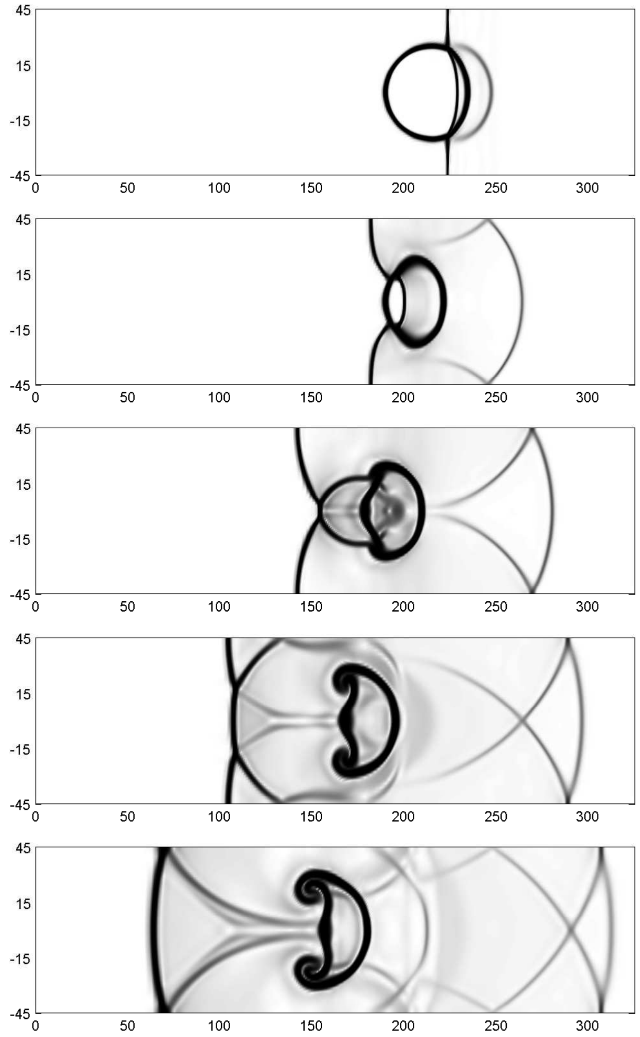

Figure 17: Same as Fig. 16 except for the schlieren images of the density.

Table 5: Example 6.9: The estimated CPU times (seconds).

Algorithm	Mesh number	CPU time
Moving mesh	260×72	1011.1
Fixed mesh	650×180	1559.6

mesh method is efficient for those two shock-bubble interaction problems, obtaining the same resolution with a much smaller number of mesh points than the uniform mesh method.

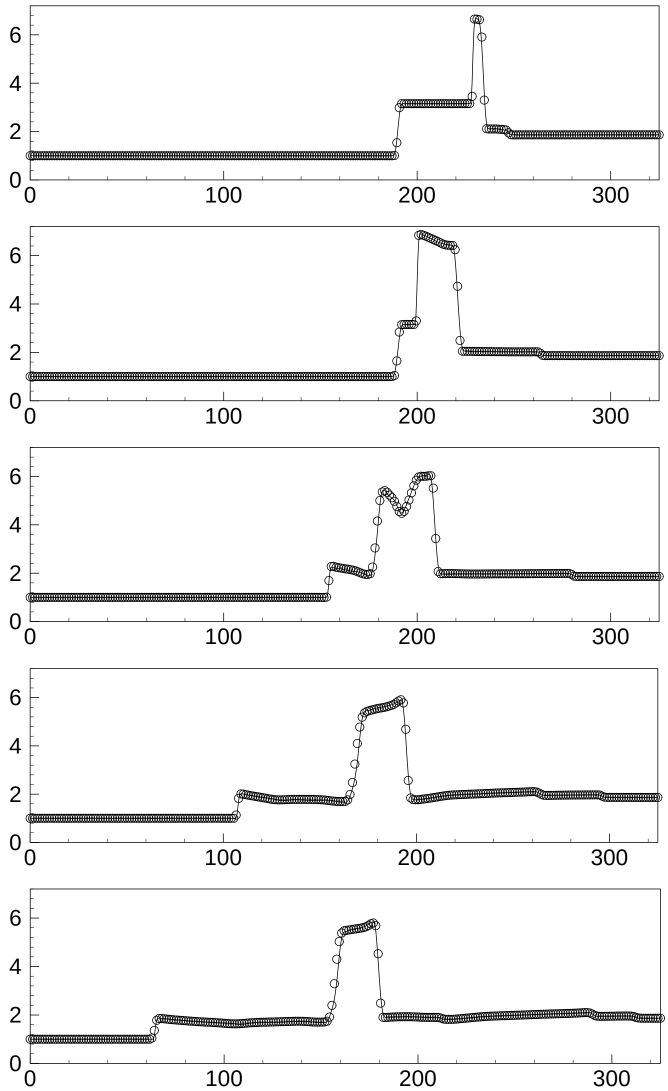

Figure 18: Same as Fig. 16 except for the densities along $y=0$. The solid line and the symbols "$\circ$" denote the computed solutions obtained on a 650×180 uniform mesh and a 260×72 moving mesh, respectively.

7 Conclusions

The paper has developed a high resolution adaptive moving mesh algorithm for two-dimensional (2D) relativistic hydrodynamics. It is formed into two "independent" parts: the time evolution of the relativistic hydrodynamic (RHD) equations and the (static) mesh iteration redistribution. In the first part, the RHD equations are solved by using a high resolution finite volume scheme on the fixed non-uniform meshes without the use of the full characteristic decomposition of the governing equations. In the second part, the mesh points are first iteratively and adaptively redistributed by the suitably chosen monitor function, and then the cell averages of the conservative variables are remapped onto the new mesh in a conservative way.

Several numerical experiments have been conducted to demonstrate robustness and efficiency of the proposed method. They are 2D smooth problem for the accuracy test, quasi 2D RHD Riemann problems, relativistic double Mach reflection, 2D RHD Riemann problems, and 2D RHD shock-bubble interaction problems. The computations demonstrate that our methods are efficient for solving problems with shock discontinuities, obtaining the same resolution with a much smaller number of grid points than the uniform mesh approach.

Appendix

This appendix derives the eigenvalues and eigenvectors of the Jacobian matrix $\partial \boldsymbol{F}_i/\partial \boldsymbol{U}$ for the two-dimensional relativistic hydrodynamic equations, where $\boldsymbol{U}$ and $\boldsymbol{F}_i$ are the vector of the conservative variables and the flux vector in x_i-direction, respectively, see Section 2, and $i=1,2$.

Denote $\mathcal{I}$ the 4×4 identity matrix and

$$\mathcal{A}_i := \frac{\partial \boldsymbol{F}_i}{\partial \boldsymbol{V}}, \quad \mathcal{A}_0 := \frac{\partial \boldsymbol{U}}{\partial \boldsymbol{V}}, \quad \mathcal{B}_i := \frac{\partial \boldsymbol{F}_i}{\partial \boldsymbol{U}}, \quad i=1,2,$$

where $\boldsymbol{V}=(\rho,v_1,v_2,p)^T$ denotes the vector of the primitive variables. Obviously, we have

$$\mathcal{B}_i = \frac{\partial \boldsymbol{F}_i}{\partial \boldsymbol{U}} = \frac{\partial \boldsymbol{F}_i}{\partial \boldsymbol{V}} \cdot \frac{\partial \boldsymbol{V}}{\partial \boldsymbol{U}} = \mathcal{A}_i \mathcal{A}_0^{-1}. \tag{A.1}$$

The eigenvalues of the Jacobian matrix $\mathcal{B}_i$ are the roots of its characteristic polynomial $\det[\mathcal{B}_i - \lambda \mathcal{I}]$, i.e., the solutions of the equation

$$0 = \det[\mathcal{B}_i - \lambda \mathcal{I}] = \frac{\det[\mathcal{D}_i(\lambda)]}{\det(\mathcal{A}_0)},$$

where $\mathcal{D}_i(\lambda) := \mathcal{A}_i - \lambda \mathcal{A}_0$. Note that the matrix $\mathcal{A}_0$ is nonsingular. Solving the equation $\det[\mathcal{D}_i(\lambda)]=0$ can conveniently give four real eigenvalues $\{\lambda_i^{(\ell)}, \ell=1,2,3,4\}$ of the Jacobian matrix $\mathcal{B}_i$ as shown in Section 2.

Now let $\lambda_i^{(\ell)}$ be one of the ℓth eigenvalue of the matrix $\mathcal{B}_i$ and $\boldsymbol{r}_i^{(\ell)}$ the vector satisfying $\mathcal{D}_i(\lambda_i^{(\ell)})\boldsymbol{r}_i^{(\ell)}=0$, i.e.,

$$0 = \mathcal{D}_i(\lambda_i^{(\ell)})\boldsymbol{r}_i^{(\ell)} = [\mathcal{A}_i\mathcal{A}_0^{-1} - \lambda_i^{(\ell)}\mathcal{I}]\mathcal{A}_0\boldsymbol{r}_i^{(\ell)} = [\mathcal{B}_i - \lambda_i^{(\ell)}\mathcal{I}]\mathcal{A}_0\boldsymbol{r}_i^{(\ell)}. \tag{A.2}$$

So $\mathcal{A}_0\boldsymbol{r}_i^{(\ell)}$ is the ℓth eigenvector of the $\mathcal{B}_i$ corresponding to the ℓth eigenvalue $\lambda_i^{(\ell)}$.

A complete set of the solution vectors to the problem $\mathcal{D}_1(\lambda_1^{(\ell)})\boldsymbol{r}_1^{(\ell)}=0$ is as follows:

$$\boldsymbol{r}_1^{(2)}=(1,0,0,0)^T,\qquad \boldsymbol{r}_1^{(3)}=(0,0,1,0)^T,$$

and

$$\boldsymbol{r}_1^{(\ell)}=\begin{pmatrix}1\\(1-\lambda_1^{(\ell)}v_1)v^{(\ell)}\\-\lambda_1^{(\ell)}v_2v^{(\ell)}\\hc_s^2\end{pmatrix},\quad \ell=1 \text{ or } 4,$$

where

$$v^{(\ell)}:=\frac{(v_1-\lambda_1^{(\ell)})}{\rho\sigma^{(\ell)}},\qquad \sigma^{(\ell)}:=-v^2\gamma^2(\lambda_1^{(\ell)})^2+2v_1\gamma^2\lambda_1^{(\ell)}-1-v_1^2\gamma^2.$$

Here $v^2=v_1^2+v_2^2$ and $\sigma^{(\ell)}$ is always non-zero if $\lambda_1^{(\ell)}$ is a real number.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (No. 10925101, 10828101), the Program for New Century Excellent Talents in University (NCET-07-0022), and the Doctoral Program of Education Ministry of China (No. 20070001036).

References

[1] D. S. Balsara, Riemann solver for relativistic hydrodynamics, J. Comput. Phys., 114 (1994), 284–297.

[2] J. U. Brackbill, An adaptive grid with directional control, J. Comput. Phys., 108 (1993), 38–50.

[3] J. U. Brackbill and J. S. Saltzman, Adaptive zoning for singular problems in two dimensions, J. Comput. Phys., 46 (1982), 342–368.

[4] C. J. Budd, W. Huang and R. D. Russell, Adaptivity with moving grids, Acta Numer., 18 (2009), 111–241.

[5] W. M. Cao, W. Z. Huang and R. D. Russell, An *r*-adaptive finite element method based upon moving mesh PDEs, J. Comput. Phys., 149 (1999), 221–244.

[6] H. D. Ceniceros and T. Y. Hou, An efficient dynamically adaptive mesh for potentially singular solutions, J. Comput. Phys., 172 (2001), 609–639.

[7] G. X. Chen, H. Z. Tang and P. W. Zhang, Second-order accurate Godunov scheme for multicomponent flows on moving triangular meshes, J. Sci. Comput., 34(1) (2008), 64–86.

[8] W. Dai and P. R. Woodward, An iterative Riemann solver for relativistic hydrodynamics, SIAM J. Sci. Stat. Comput., 18 (1997), 982–995.

[9] S. F. Davis and J. E. Flaherty, An adaptive finite element method for initial-boundary value problems for partial differential equations, SIAM J. Sci. Stat. Comput., 3 (1982), 6–27.

[10] Y. N. Di, R. Li, T. Tang and P. W. Zhang, Moving mesh finite element methods for the incompressible Navier-Stokes equations, SIAM J. Sci. Comput., 26 (2005), 1036–1056.

[11] A. Dolezal and S. S. M. Wong, Relativistic hydrodynamics and essentially non-oscillatory shock capturing schemes, J. Comput. Phys., 120 (1995), 266–277.

[12] R. Donat, J. A. Font, J. M. Ibáñez and A. Marquina, A flux-split algorithm applied to relativistic flows, J. Comput. Phys., 146 (1998), 58–81.

[13] G. C. Duncan and P. A. Hughes, Simulations of relativistic extragalactic jets, Astrophys. J., 436 (1994), L119–L122.

[14] A. S. Dvinsky, Adaptive grid generation from harmonic maps on Riemannian manifolds, J. Comput. Phys., 95 (1991), 450–476.

[15] F. Eulderink and G. Mellema, General relativistic hydrodynamics with a Roe solver, Astron. Astrophys., Suppl., 110 (1995), 587–623.

[16] S. A. E. G. Falle and S. S. Komissarov, An upwind numerical scheme for relativistic hydrodynamics with a general equation of state, Mon. Not. R. Astron. Soc., 278 (1996), 586–602.

[17] J. A. Font, J. M. Ibáñez, A. Marquina and J. M. Martí, Multidimensional relativistic hydrodynamics: characteristic fields and modern high-resolution shock-capturing schemes, Astron. Astrophys., 282(1) (1994), 304–314.

[18] E. Han, J. Q. Li and H. Z. Tang, Accuracy of the adaptive GRP scheme and the simulation of 2-D Riemann problems for compressible Euler equations, Commun. Comput. Phys., 10 (2011), 577–606.

[19] J. Q. Han and H. Z. Tang, An adaptive moving mesh method for multidimensional ideal magnetohydrodynamics, J. Comput. Phys., 220 (2007), 791–812.

[20] L. D. Landau and E. M. Lifshitz, Fluid Mechanics, Butterworth-Heinemann, second edition, 1987.

[21] R. J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.

[22] R. Li, T. Tang and P. W. Zhang, Moving mesh methods in multiple dimensions based on harmonic maps, J. Comput. Phys., 170 (2001), 562–588.

[23] R. Li, T. Tang and P. W. Zhang, A moving mesh finite element algorithm for singular problems in two and three space dimensions, J. Comput. Phys., 177 (2002), 365–393.

[24] S. Li and L. Petzold, Moving mesh methods with upwinding schemes for time-dependent PDEs, J. Comput. Phys., 131 (1997), 368–377.

[25] A. Lucas-Serrano, J. A. Font, J. M. Ibáñez and J. M. Martí, Assessment of a high-resolution central scheme for the solution of the relativistic hydrodynamics equations, Astron. Astrophys., 428(2) (2004), 703–715.

[26] J. M. Martí and E. Müller, The analytical solution of the Riemann problem in relativistic hydrodynamics, J. Fluid Mech., 258(1) (1994), 317–333.

[27] J. M. Martí and E. Müller, Numerical hydrodynamics in special relativity, Living Rev. Relativ., 6(7) (2003), 1–100.

[28] A. Mignone and 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 and G. Bodo, The piecewise parabolic method for multidimensional relativistic fluid dynamics, Astrophys. J., Suppl., 160(1) (2005), 199–219.

[30] K. Miller and R. N. Miller, Moving finite element I, SIAM J. Numer. Anal., 18 (1981), 1019–1032.

[31] W. Q. Ren and X. P. Wang, An iterative grid redistribution method for singular problems in multiple dimensions, J. Comput. Phys., 159 (2000), 246–273.

[32] V. Schneider, U. Katscher, D. H. Rischke, B. Waldhauser, J. A. Maruhn and C. D. Munz, New algorithms for ultra-relativistic numerical hydrodynamics, J. Comput. Phys., 105 (1993), 92–107.

[33] J. M. Stockie, J. A. Mackenzie and R. D. Russell, A moving mesh method for one-dimensional hyperbolic conservation laws, SIAM J. Sci. Comput., 22 (2001), 1791–1813.

[34] Z. J. Tan, Z. R. Zhang, Y. Q. Huang and T. Tang, Moving mesh methods with locally varying time steps, J. Comput. Phys., 200 (2004), 347–367.

[35] H. Z. Tang, A moving mesh method for the Euler flow calculations using a directional monitor function, Commun. Comput. Phys., 1 (2006), 656–676.

[36] H. Z. Tang and T. Tang, Adaptive mesh methods for one- and two-dimensional hyperbolic conservations laws, SIAM J. Numer. Anal., 41 (2003), 487–515.

[37] H. Z. Tang, T. Tang and P. W. Zhang, An adaptive mesh redistribution method for nonliear Hamilton-Jacobi equations in two- and three-dimensions, J. Comput. Phys., 188 (2003), 543–572.

[38] T. Tang, Moving mesh methods for computational fluid dynamics, Contemp. Math., 383 (2005), 141–174.

[39] A. van Dam and P. A. Zegeling, Balanced monitoring of flow phenomena in moving mesh methods, Commun. Comput. Phys., 7 (2010), 138–170.

[40] B. van Leer, Towards the ultimate conservative difference scheme III. upstream-centered finite-difference schemes for ideal compressible flow, J. Comput. Phys., 23 (1977), 263–299.

[41] D. S. Wang and X. P. Wang, Three dimensional adaptive method based on iterative grid redistribution, J. Comput. Phys., 199 (2004), 423–436.

[42] J. R. Wilson, Numerical study of fluid flow in a Kerr space, Astrophys. J., 173 (1972), 431–438.

[43] J. R. Wilson, A numerical method for relativistic hydrodynamics, in L. L. Smarr, editor, Sources of Gravitational Radiation, 423–446, Cambridge University Press, 1979.

[44] A. M. Winslow, Numerical solution of the quasi-linear Poisson equation in a nonuniform triangle mesh, J. Comput. Phys., 1 (1966), 149–172.

[45] L. D. Zanna and N. Bucciantini, An efficient shock-capturing central-type scheme for multidimensional relativistic flows, I. hydrodynamics, Astron. Astrophys., 390(3) (2002), 1177–1186.

[46] W. Q. Zhang and A. I. MacFadyen, RAM: a relativistic adaptive mesh refinement hydrodynamics code, Astrophys. J. Suppl., 164(1) (2006), 255–279.