Journal of Computational Physics 229 (2010) 1448–1466

Contents lists available at ScienceDirect

Journal of Computational Physics

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

An adaptive GRP scheme for compressible fluid flows

Ee Han [a], Jiequan Li [a,*], Huazhong Tang [b]

[a] School of Mathematical Science, Capital Normal University, Beijing 100048, PR China
[b] CAPT and LMAM, School of Mathematical Sciences, Peking University, Beijing 100871, PR China

ARTICLE INFO

Article history:
Received 2 March 2009
Received in revised form 9 October 2009
Accepted 14 October 2009
Available online 31 October 2009

MSC:
65M06
76M12
35L60

Keywords:
GRP scheme
Adaptive moving mesh method
Monitor function
Conservative interpolation

ABSTRACT

This paper presents a second-order accurate adaptive generalized Riemann problem (GRP) scheme for one and two dimensional compressible fluid flows. The current scheme consists of two independent parts: Mesh redistribution and PDE evolution. The first part is an iterative procedure. In each iteration, mesh points are first redistributed, and then a conservative interpolation formula is used to calculate the cell-averages and the slopes of conservative variables on the resulting new mesh. The second part is to evolve the compressible fluid flows on a fixed nonuniform mesh with the Eulerian GRP scheme, which is directly extended to two-dimensional arbitrary quadrilateral meshes. Several numerical examples show that the current adaptive GRP scheme does not only improve the resolution as well as accuracy of numerical solutions with a few mesh points, but also reduces possible errors or oscillations effectively.

1. Introduction

The generalized Riemann problem (GRP) scheme, as an analytic second order accurate extension of the Godunov scheme, was originally developed for one-dimensional (1D) system of an unsteady and inviscid flows [1,3]. The basic idea of the GRP scheme consists of replacing the exact solution by a piecewise linear function and analytically solving a generalized Riemann problem at each cell interface to yield numerical fluxes. Recently a direct Eulerian GRP scheme was presented in [4,5,20–22], aiming at getting rid of the auxiliary Lagrangian scheme, which was essential in [1], and solving the one-dimensional (1D) generalized Riemann problem directly in the Eulerian coordinates by employing the regularity property of the Riemann invariants.

The direct Eulerian GRP scheme is efficient and robust in capturing hydrodynamic singularities (shocks and contact discontinuities) for most cases. In the meantime it also inherits drawbacks from many Godunov-type schemes, such as the instability of stationary shocks and start-up errors in a single shock wave of the one-dimensional compressible fluid flows. For two-dimensional cases, the GRP scheme was extended by employing the Strang splitting method in [5], but it had to be restricted on uniform rectangular meshes. The purpose of this paper is to develop one-dimensional and two-dimensional adaptive GRP schemes by combining the Eulerian GRP scheme [5] with the adaptive moving mesh method [31]. With such efforts, the afore mentioned drawbacks or restrictions are totally overcome.

Our present adaptive GRP scheme consists of two independent parts: Evolution of PDEs with the GRP scheme on an arbitrary quadrangular mesh and the mesh redistribution with the Gauss–Seidel iteration method. In this context one key

* Corresponding author. Tel.: +86 10 68905511x407; fax: +86 10 68980951.
E-mail addresses: eehan84@yahoo.com (E. Han), jiequan@mail.cnu.edu.cn (J. Li), hztang@math.pku.edu.cn (H. Tang).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.10.038

ingredient is to extend the Eulerian GRP scheme over arbitrary quadrangular meshes. The reason is that the adaptive moving mesh method can locally cluster or spread out mesh points according to the solution variations, and that this method will totally break the uniform mesh distribution. Note that it makes sense in its own to develop a two-dimensional or multi-dimensional GRP scheme on arbitrary meshes. Our approach of the two-dimensional GRP scheme on arbitrary quadrangle meshes is to integrate the conservation laws over an hexahedral control volume in the space time domain, and then use the average of conservative variables (mass, momentum, and energy) to replace their integrals on the top and bottom faces. Numerical fluxes on four lateral faces are approximated by replacing true solutions with the centroid point values, which are analytically obtained by solving the associated generalized Riemann problem in the time direction. This generalized Riemann problem is defined in the outward normal direction of each quadrangle boundary.

Another key ingredient in the present scheme is to define the primitive slope after every step of the mesh iteration distribution, since the primitive slope cannot be directly obtained on the resulting new mesh. The conservative interpolation developed in [31] is used to solve this problem. Some efficient monitor functions are discussed and summarized in numerical experiments. From the numerical results in Section 5, we can see that the adaptive GRP scheme will not only improve the resolution as well as the accuracy of numerical solutions with much fewer meshes, but also effectively reduce possible errors (oscillations), which may be present in many Godunov-type schemes for the compressible fluid flows [17]. That is to say, the present adaptive GRP scheme provides an efficient and robust way to achieve very accurate solutions, thus it would be useful in practical applications.

This paper is organized as follows. In Section 2, the direct Eulerian GRP scheme for two-dimensional planar compressible fluid flows is reviewed and extended to arbitrary quadrangular meshes for general two-dimensional cases. In Section 3, the adaptive moving mesh method is illustrated, in particular, the method how to remap the primitive slope. Section 4 shows the algorithm of the adaptive GRP scheme. Numerical experiments are carried out in Section 5 to display the performance of this scheme. A final conclusion is given in Section 6.

2. The GRP scheme in one and two-dimensions

In this section we first review the GRP scheme for two-dimensional (2D) planar compressible fluid flows, and then extend it for general two-dimensional cases on arbitrary quadrangular meshes. This serves to develop our adaptive GRP scheme, although it has its own significance.

2.1. Review of 2D planar GRP scheme

The following system is used to describe 2D planar compressible fluid flows,

$$\frac{\partial U}{\partial t}+\frac{\partial F(U)}{\partial x}=0,\quad U=\begin{pmatrix}\rho\\ \rho u\\ \rho vE\end{pmatrix},\quad F(U)=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ uE+pu\end{pmatrix}, \tag{2.1}$$

where ρ, e are the density and the internal energy, respectively, $p=p(\rho,e)$ is the pressure, u is the velocity along the x-direction and v is the velocity perpendicular to the x-direction, $E=\rho e+\rho\frac{u^2+v^2}{2}$ is the total energy.

Denote spatial grid points with $\{x_j; j=1,\ldots,J\}$, and interface points $x_{j+\frac{1}{2}}=(x_j+x_{j+1})/2$. Define the cells $C_j=\left[x_{j-\frac{1}{2}},x_{j+\frac{1}{2}}\right]$ with length $\Delta x_j=x_{j+\frac{1}{2}}-x_{j-\frac{1}{2}}$. Let U_j^n be the average value of U over the cell C_j at time $t_n=n\Delta t$, and assume that the data at time $t=t_n$ are piecewise linear with a slope σ_j^n,

$$U(x,t_n)=U_j^n+\sigma_j^n(x-x_j),\qquad x\in\left(x_{j-\frac{1}{2}},x_{j+\frac{1}{2}}\right). \tag{2.2}$$

Then the GRP scheme consists of the following three steps to define a pair of unknown vectors $(U_j^{n+1},\sigma_j^{n+1})$.

Step 1. Given (2.2), calculate mid-point values $U_{j+\frac{1}{2}}^{n+\frac{1}{2}}$ approximately with the formula

$$U_{j+\frac{1}{2}}^{n+\frac{1}{2}}=U_{j+\frac{1}{2}}^{n}+\frac{\Delta t}{2}\left(\frac{\partial U}{\partial t}\right)_{j+\frac{1}{2}}^{n}, \tag{2.3}$$

where $U_{j+\frac{1}{2}}^n$ and $\left(\frac{\partial U}{\partial t}\right)_{j+\frac{1}{2}}^n$ are defined via solving the local generalized Riemann problem (GRP) at $\left(x_{j+\frac{1}{2}},t_n\right)$, and specified in Section (2.2).

Step 2. Evaluate the interior cell averages by using the updating formula

$$U_j^{n+1}=U_j^n-\frac{\Delta t}{\Delta x_j}\left(F\left(U_{j+\frac{1}{2}}^{n+\frac{1}{2}}\right)-F\left(U_{j-\frac{1}{2}}^{n+\frac{1}{2}}\right)\right),\quad j=1,\ldots,J. \tag{2.4}$$

Step 3. Update the slope σ_j^{n+1} by the following procedure,

$$U_{j+\frac{1}{2}}^{n+1,-} = U_{j+\frac{1}{2}}^{n} + \Delta t\left(\frac{\partial U}{\partial t}\right)_{j+\frac{1}{2}}^{n},$$
$$\sigma_j^{n+1,-} = \frac{1}{\Delta x_j}\left(U_{j+\frac{1}{2}}^{n+1,-} - U_{j-\frac{1}{2}}^{n+1,-}\right). \tag{2.5}$$

In order to suppress local oscillations near discontinuities, a monotonicity algorithm-slope limiter is applied, which is

$$\sigma_j^{n+1} = \text{minmod}\left(\alpha\frac{U_j^{n+1} - U_{j-1}^{n+1}}{\Delta x_j}, \sigma_j^{n+1,-}, \alpha\frac{U_{j+1}^{n+1} - U_j^{n+1}}{\Delta x_j}\right), \tag{2.6}$$

where the parameter $\alpha \in [0, 2)$.

It is evident that the main task of the GRP scheme is to evaluate $U_{j+\frac{1}{2}}^n$ and $(\frac{\partial U}{\partial t})_{j+\frac{1}{2}}^n$. The former can be obtained with the standard Riemann solver, while the latter is achieved by analytically solving the generalized Riemann problem. The detail will be given in the next subsection. This is the crucial part of the GRP scheme. It is just this point that it is different from the MUSCL scheme [34].

2.2. The generalized Riemann problem

Now the generalized Riemann problem is solved at $\left(x_{j+\frac{1}{2}}, t_n\right)$ to define the vector-values $U_{j+\frac{1}{2}}^n$ and $(\frac{\partial U}{\partial t})_{j+\frac{1}{2}}^n$. We shift $\left(x_{j+\frac{1}{2}}, t_n\right)$ to (0,0) and denote $U_L = U_j^n + \frac{\Delta x_j}{2}\sigma_j^n, U_R = U_{j+1}^n - \frac{\Delta x_{j+1}}{2}\sigma_{j+1}^n, U_L' = \sigma_j^n$ and $U_R' = \sigma_{j+1}^n$. Then locally we have the generalized Riemann problem for (2.1) with the piecewise linear initial data

$$U(x,0) = \begin{cases} U_L + xU_L', & x < 0, \\ U_R + xU_R', & x > 0. \end{cases} \tag{2.7}$$

The solution of (2.1) with the uniform constant data (i.e. $U_L' = U_R' \equiv 0$ in (2.7) is referred to as *the Riemann solution*, determining the local wave configuration of (2.1) and (2.7) around (0,0). The Riemann solution can be obtained via a Riemann solver [3,33], being denoted here by $R^A(x/t; U_L, U_R)$. Denote also by $U(x,t)$ the solution of (2.1) and (2.7). Then we can define $U_{j+1/2}^n$ as

$$U_{j+1/2}^n := \lim_{t\to 0^+} U(0,t) = R^A(0; U_L, U_R). \tag{2.8}$$

As for $(\frac{\partial U}{\partial t})_{j+1/2}^n$, it is analytically defined through the four physical components (ρ, u, v, p). Since the flow variables u and p are continuous across the contact discontinuity in the intermediate region, we can first solve a pair of linear algebraic equations to obtain $(\frac{\partial p}{\partial t})_{j+1/2}^n$ and $(\frac{\partial u}{\partial t})_{j+1/2}^n$,

$$\begin{cases} a_L\left(\frac{\partial p}{\partial t}\right)_{j+1/2}^n + b_L\left(\frac{\partial u}{\partial t}\right)_{j+1/2}^n = d_L, \\ a_R\left(\frac{\partial p}{\partial t}\right)_{j+1/2}^n + b_R\left(\frac{\partial u}{\partial t}\right)_{j+1/2}^n = d_R. \end{cases} \tag{2.9}$$

Then $(\frac{\partial \rho}{\partial t})_{j+1/2}^n$ and $(\frac{\partial v}{\partial t})_{j+1/2}^n$ are obtained successively, depending on the location of the contact discontinuity. Here (a_L, b_L, d_L) and (a_R, b_R, d_R) can be expressed explicitly. We refer to [4,5] for details.

2.3. The 2D GRP scheme on arbitrary quadrangular meshes

The two-dimensional system of an unsteady and inviscid compressible flow is written as:

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} + \frac{\partial G(U)}{\partial y} = 0, \quad U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \quad F(U) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix}, \quad G(U) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}. \tag{2.10}$$

Given a partition of the physical domain Ω_p, denoted by $\{A_{i,j}; i,j \in \mathbb{Z}\}$, where $A_{i,j}$ is an arbitrary quadrangle with four vertices $(x_{i+p,j+q}, y_{i+p,j+q})$, $p, q = \pm\frac{1}{2}$. For simplicity, we fix (i,j), and denote four vertexes by $\boldsymbol{x}^k = (x^k, y^k), k = 1,2,3,4$, so that $\boldsymbol{x}^1 = (x^1, y^1) = \left(x_{i-\frac{1}{2},j-\frac{1}{2}}, y_{i-\frac{1}{2},j-\frac{1}{2}}\right)$. They are further ordered in an counter-clockwise manner.

Let $C_{i,j} = A_{i,j} \times [t_n, t_{n+1})$ be a control volume. The four lateral faces are denoted by S_k, and $A_{i,j}^k$ is the kth neighboring quadrangle of $A_{i,j}, k = 1,2,3,4$. S_k and $A_{i,j}^k$ have a common boundary l_k with $A_{i,j}$, as shown in Fig. 2.1.

Let $U_{i,j}^n$ be the average value of the U over $A_{i,j}$ at the time t_n, and assume that the data at time $t = t_n$ are piecewise linear function

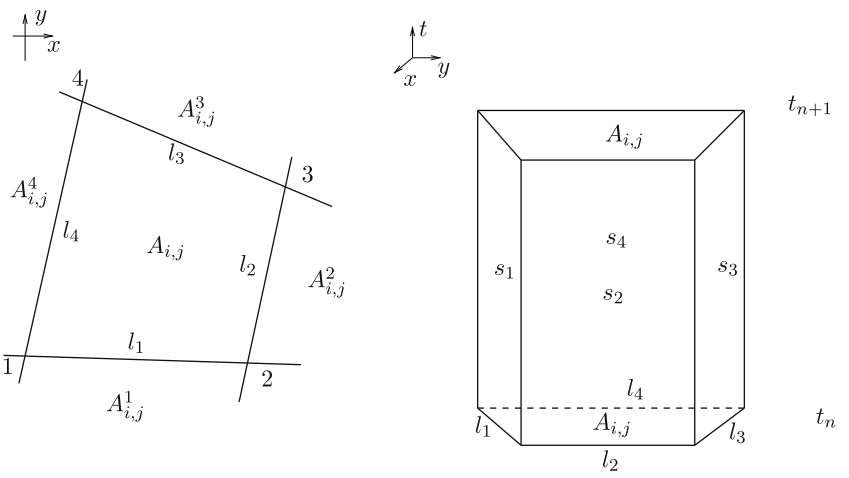


Fig. 2.1. Quadrangle meshes and hexahedral control volumes.

$$U_{A_{ij}}(x,y,t_n) = U^n_{ij} + (x - x_{ij})(\sigma_x)^n_{ij} + (y - y_{ij})(\sigma_y)^n_{ij}, \quad (x,y) \in A_{ij}; \tag{2.11}$$

where (x_{ij}, y_{ij}) is the centroid of A_{ij}. Integrating (2.10) over C_{ij} and using the divergence theorem, we obtain:

$$|A_{ij}|U^{n+1}_{ij} = |A_{ij}|U^n_{ij} - \sum_{k=1}^{4} \int_{S_k} [F(U)\mu_k + G(U)\nu_k]ds, \tag{2.12}$$

where (μ_k, ν_k) is the unit outward normal of the boundary $l_k, k = 1,2,3,4$, and $|A_{ij}|$ is the area of A_{ij},

$$|A_{ij}| = \frac{1}{2}[(x^3 - x^1)(y^4 - y^2) - (x^4 - x^2)(y^3 - y^1)].$$

Using the centroid values of four lateral faces to approximate the solution $U(x,y,t)$ in the flux integral of (2.12), we formally obtain

$$U^{n+1}_{ij} = U^n_{ij} - \frac{1}{|A_{ij}|}\sum_{k=1}^{4} Q_k, \tag{2.13}$$

where the numerical flux function Q_k is

$$Q_k = \left[F\left(U^{n+\frac{1}{2}}_{S_k}\right)\mu_k + G\left(U^{n+\frac{1}{2}}_{S_k}\right)\nu_k\right]|S_k|, \tag{2.14}$$

and $|S_k| = |l_k|\Delta t$ is the area of S_k, $|l_k|$ denotes the length of the edge $l_k, k = 1,2,3,4$. The centroid values $U^{n+\frac{1}{2}}_{S_k}$ are computed from the following generalized Riemann problem. Define the function $H(U;\mu_k,\nu_k) = F(U)\mu_k + G(U)\nu_k$ and $\xi = x\mu_k + y\nu_k$, where (x,y) are shifted from $\left(x^{\bar{k}}, y^{\bar{k}}\right)$ to (0,0) and $k = 1,2,3,4$, then the local generalized Riemann problem is defined as

$$\frac{\partial U}{\partial t} + \frac{\partial H(U;\mu_k,\nu_k)}{\partial \xi} = 0, \quad U(\xi,0) = \begin{cases} U_{L,k} + \xi U'_{L,k}, & \xi < 0, \\ U_{R,k} + \xi U'_{R,k}, & \xi > 0, \end{cases} \tag{2.15}$$

where

$$\begin{cases} U_{L,k} = U_{A_{ij}}\left(x^{\bar{k}}, y^{\bar{k}}, t^n\right), \\ U_{R,k} = U_{A^k_{ij}}\left(x^{\bar{k}}, y^{\bar{k}}, t^n\right), \end{cases} \quad \begin{cases} U'_{L,k} = (\sigma_x)^n_{i,j}\mu_k + (\sigma_y)^n_{ij}\nu_k, \\ U'_{R,k} = (\sigma_x)^n_{A^k_{ij}}\mu_k + (\sigma_y)^n_{A^k_{ij}}\nu_k. \end{cases} \tag{2.16}$$

and $\left(x^{\bar{k}}, y^{\bar{k}}\right)$ is the middle point of the boundary segment l_k. This generalized Riemann problem is solved in the exactly same way as the planar case (2.1) and (2.7). Thus the algorithm of 2D GRP scheme based on arbitrary quadrangle meshes can be described as follows, parallel to the planar case.

Step 1. Given the piecewise initial data (2.11), calculate the centroid point values $U^{n+\frac{1}{2}}_{S_k} k = 1,2,3,4$ for every cell A_{ij} with the formula

$$U^{n+\frac{1}{2}}_{S_k} = U^n_{S_k} + \frac{\Delta t}{2}\left(\frac{\partial U}{\partial t}\right)^n_{S_k}, \tag{2.17}$$

where $U^n_{S_k}$ and $\left(\frac{\partial U}{\partial t}\right)^n_{S_k}$ are obtained from the generalized Riemann problem (2.15) and (2.16).

Step 2. Evaluate the new cell averages $U_{i,j}^{n+1}$ using Eq. (2.13).

Step 3. Update the slopes $(\sigma_z)_{i,j}^{n+1}$ (z represents x or y) from the following approximate procedure. Define

$$U_{S_k}^{n+1,-} := U_{S_k}^{n} + \Delta t \left(\frac{\partial U}{\partial t}\right)_{S_k}^{n}, \quad k = 1,2,3,4,$$

and

$$\sigma_q^{n+1,-} := \begin{cases} \frac{1}{|\boldsymbol{x}^3\boldsymbol{x}^1|}\left(U_{S_3}^{n+1,-} - U_{S_1}^{n+1,-}\right), & q = 31, \\ \frac{1}{|\boldsymbol{x}^4\boldsymbol{x}^2|}\left(U_{S_4}^{n+1,-} - U_{S_2}^{n+1,-}\right), & q = 42. \end{cases} \tag{2.18}$$

Calculate the slopes in $\vec{\boldsymbol{x}^3\boldsymbol{x}^1}$ and $\vec{\boldsymbol{x}^4\boldsymbol{x}^2}$ directions,

$$(\sigma_q)_{i,j}^{n+1} = \begin{cases} \text{minmod}\left(\alpha\frac{U_{i,j+1}^{n+1} - U_{i,j}^{n+1}}{|\boldsymbol{x}_{i,j+1}\boldsymbol{x}_{i,j}|}, \sigma_q^{n+1,-}, \alpha\frac{U_{i,j}^{n+1} - U_{i,j-1}^{n+1}}{|\boldsymbol{x}_{i,j}\boldsymbol{x}_{i,j-1}|}\right), & q = 31, \\ \text{minmod}\left(\alpha\frac{U_{i+1,j}^{n+1} - U_{i,j}^{n+1}}{|\boldsymbol{x}_{i+1,j}\boldsymbol{x}_{i,j}|}, \sigma_q^{n+1,-}, \alpha\frac{U_{i,j}^{n+1} - U_{i-1,j}^{n+1}}{|\boldsymbol{x}_{i,j}\boldsymbol{x}_{i-1,j}|}\right), & q = 42, \end{cases} \tag{2.19}$$

where $|\boldsymbol{x}_{i,j}\boldsymbol{x}_{i,j-1}| = \sqrt{(x_{i,j} - x_{i-1,j})^2 + (y_{i,j} - y_{i-1,j})^2}$ and $\alpha \in [0,2)$. Then $(\sigma_z)_{i,j}^{n+1}$ (z represents x or y) are obtained from $(\sigma_q)_{i,j}^{n+1}$ ($q = 42$ and 31) by using the local coordinate transformation similar to that in (4.1).

3. Adaptive mesh redistribution

This section illustrates the adaptive mesh redistribution method, in particular, the method of re-mapping the primitive slopes and the choice of the monitor function. This adaptive moving mesh method has become one of the important approaches to resolve dynamically singular or nearly singular solutions in fairly localized regions, such as those around shock waves etc. In the past several decades, there were a lot of important progresses in moving mesh methods for partial differential equations, including the variational approach of Winslow [37], Brackbill et al. [6,7], Dvinsky [14], and Li et al. [13,24,25]; moving finite element methods of Miller and Miller [27] and Davis and Flaherty [12]; and moving mesh PDEs of Cao et al. [8,9]. Some recent works on the moving mesh methods can be found in [10,18,16,32,11,35,36] and references therein.

3.1. Mesh redistributions based on a variational method

The adaptive mesh can be regarded as an approximation of a coordinate transformation between computational coordinates $\xi \in \Omega_c$ (with uniform mesh partition) and physical coordinates $\boldsymbol{x} \in \Omega_p$ (with nonuniform adaptive mesh). Here $\boldsymbol{\xi} = \xi, \boldsymbol{x} = x$ for 1D cases, and $\boldsymbol{\xi} = (\xi,\eta), \boldsymbol{x} = (x,y)$ for 2D cases.

Let $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$ be the coordinate transformation from the computation domain to the physical domain, and $\boldsymbol{\xi} = \boldsymbol{\xi}(\boldsymbol{x})$ denote its inversion. In a variational approach, a "mesh-energy" functional defined in the computational domain is

$$E(\boldsymbol{x}) = \frac{1}{2}\sum_{k=1}^{d}\int_{\Omega_c} \nabla^\top x_k G_k \nabla x_k d\boldsymbol{\xi} \tag{3.1}$$

where d is the number of space dimension, $\nabla = (\partial_{\xi_1}, \ldots, \partial_{\xi_d})^\top$, G_k are given symmetric positive definite matrices depending on the underlying solution to be adapted. In particular, $G_k = \omega I$ are often used, where I is the identity matrix, and the monitor function ω is a positive weighted function. This produces isotropic/nondirectional mesh adaptation. The corresponding Euler–Lagrange equations of (3.1) are

$$\nabla \cdot (\omega \nabla x_k) = 0, \quad 1 \leqslant k \leqslant d. \tag{3.2}$$

In the present paper we only consider 1D and 2D cases (i.e. $d = 1$ or 2), and use the Gauss–Seidel iteration method to solve the mesh-moving Eq. (3.2).

The 1D case.

$$\begin{cases} (\omega x_\xi)_\xi = 0, & \xi \in [0,1], \\ x(0) = a, & x(1) = b. \end{cases} \tag{3.3}$$

The Gauss–Seidel iteration method takes the form

$$\omega_{j+1} x_{j+\frac{3}{2}}^{[\nu]} - (\omega_j + \omega_{j-1}) x_{j+\frac{1}{2}}^{[\nu+1]} + \omega_{j-1} x_{j-\frac{1}{2}}^{[\nu+1]} = 0, \tag{3.4}$$

where $\omega_j = \omega\left(U_j^{[\nu]}\right)$, $[\nu]$ is the iteration step. The details of the monitor function are listed in Section 3.4.

The 2D case.

$$\begin{cases} (\omega x_\xi)_\xi + (\omega x_\eta)_\eta = 0, \\ (\omega y_\xi)_\xi + (\omega y_\eta)_\eta = 0, \end{cases} \tag{3.5}$$

with the boundary condition $x(0,\eta) = a, x(1,\eta) = b, y(\xi,0) = c, y(\xi,1) = d$.

The Gauss–Seidel iteration formulae are

$$\begin{aligned} &\alpha_{i+1j+\frac{1}{2}}\left(x^{[v]}_{i+\frac{3}{2}j+\frac{1}{2}} - x^{[v+1]}_{i+\frac{1}{2}j+\frac{1}{2}}\right) - \alpha_{ij+\frac{1}{2}}\left(x^{[v+1]}_{i+\frac{1}{2}j+\frac{1}{2}} - x^{[v+1]}_{j-\frac{1}{2}j+\frac{1}{2}}\right) + \beta_{i+\frac{1}{2}j+1}\left(x^{[v]}_{i+\frac{1}{2}j+\frac{3}{2}} - x^{[v+1]}_{i+\frac{1}{2}j+\frac{1}{2}}\right) \\ &\quad - \beta_{i+\frac{1}{2}j}\left(x^{[v+1]}_{i+\frac{1}{2}j+\frac{1}{2}} - x^{[v+1]}_{i+\frac{1}{2}j-\frac{1}{2}}\right) = 0, \\ &\alpha_{i+1j+\frac{1}{2}}\left(y^{[v]}_{i+\frac{3}{2}j+\frac{1}{2}} - y^{[v+1]}_{i+\frac{1}{2}j+\frac{1}{2}}\right) - \alpha_{ij+\frac{1}{2}}\left(y^{[v+1]}_{i+\frac{1}{2}j+\frac{1}{2}} - y^{[v+1]}_{j-\frac{1}{2}j+\frac{1}{2}}\right) + \beta_{i+\frac{1}{2}j+1}\left(y^{[v]}_{i+\frac{1}{2}j+\frac{3}{2}} - y^{[v+1]}_{i+\frac{1}{2}j+\frac{1}{2}}\right) \\ &\quad - \beta_{i+\frac{1}{2}j}\left(y^{[v+1]}_{i+\frac{1}{2}j+\frac{1}{2}} - y^{[v+1]}_{i+\frac{1}{2}j-\frac{1}{2}}\right) = 0, \end{aligned} \tag{3.6}$$

where the parameters are given by

$$\alpha_{ij+\frac{1}{2}} = \frac{1}{2}\left[\omega\left(U^{[v]}_{ij}\right) + \omega\left(U^{[v]}_{ij+1}\right)\right], \quad \beta_{i+\frac{1}{2}j} = \frac{1}{2}\left[\omega\left(U^{[v]}_{ij}\right) + \omega\left(U^{[v]}_{i+1j}\right)\right].$$

3.2. Conservative interpolation on new meshes

After each iterative step of (3.5) (resp. (3.6)), we need to remap the approximate numerical solutions from old mesh C_j (resp. A_{ij}) onto the newly resulting mesh $\widetilde{C}_j$ (resp. $\widetilde{A}_{ij}$). Tang and Tang in [31] proposed a conservative interpolation method for one and two-dimensional conservation laws. It is an efficient interpolation method in preserving conservation of the mass at each mesh redistribution step. Let $\widetilde{U}_j$(resp. $\widetilde{U}_{ij}$) be the new cell average of the 1D conservative variables (resp. 2D).

The conservative interpolation for the 1D case is

$$|\Delta\tilde{x}_j|\widetilde{U}_j = |\Delta x_j|U_j - \left((cU)_{j+\frac{1}{2}} - (cU)_{j-\frac{1}{2}}\right), \tag{3.7}$$

where $c(x) = x - \tilde{x}$ and $(cU)_{j+\frac{1}{2}}$ denotes the value of cU at the interface $x_{j+\frac{1}{2}}$. The linear flux cU will be approximated by an upwind numerical flux. In general, we have

$$(cU)_{j+\frac{1}{2}} = \frac{c_{j+\frac{1}{2}} + |c_{j+\frac{1}{2}}|}{2}\left(U_j + \frac{\Delta x_j}{2}\sigma_j\right) + \frac{c_{j+\frac{1}{2}} - |c_{j+\frac{1}{2}}|}{2}\left(U_{j+1} - \frac{\Delta x_{j+1}}{2}\sigma_{j+1}\right). \tag{3.8}$$

For the 2D case, the conservation interpolation is

$$|\widetilde{A}_{ij}|\widetilde{U}_{ij} = |A_{ij}|U_{ij} - [(n_cU)_{l_4} + (n_cU)_{l_2}] - [(n_cU)_{l_3} + (n_cU)_{l_1}], \tag{3.9}$$

where $n_c = c^x\mu + c^y\nu, (c^x, c^y) = (x - \tilde{x}, y - \tilde{y}), (n_cU)_{l_k}$ denotes the value of the n_cU at the boundary l_k. In practice, we always use the following upwind approximation to define $(Un_c)_{l_k}$:

$$(n_cU)_{l_k} = \frac{(n_c)_{l_k} + |(n_c)_{l_k}|}{2}(U_{L,k}) + \frac{(n_c)_{l_k} - |(n_c)_{l_k}|}{2}(U_{R,k}), \tag{3.10}$$

where $U_{m,k}, m = L$ or R is defined similar to (2.16). The details of this conservation interpolation can be found in [30].

3.3. Primitive slope on new meshes

The primitive slope should be defined on the resulting new mesh, if the minmod limiter (2.6) (1D) or (2.19) (2D) are used to reconstruct the piecewise linear functions on it. Here we take the 1D case as an example. Let $\left\{\tilde{x}^{n+1}_j\right\}_{j=0}^{J}$ be the resulting new mesh at t_{n+1} and $\left\{x^n_j\right\}_{j=0}^{J}$ the mesh distribution at t_n, as shown in Fig. 3.1.

One simple way to define the new primitive slope is

$$\tilde{\sigma}^{n+1}_j = \frac{\widetilde{U}^{n+1}_{j+\frac{1}{2}} - \widetilde{U}^{n+1}_{j-\frac{1}{2}}}{\Delta\tilde{x}^{n+1}_j}, \tag{3.11}$$

where the interface value $\widetilde{U}^{n+1}_{j+\frac{1}{2}}$ can be calculated from the generalized Riemann problem (2.7) along the direction

$$\lambda = \frac{\tilde{x}^{n+1}_{j+\frac{1}{2}} - x^n_{j+\frac{1}{2}}}{\Delta t_n}. \tag{3.12}$$

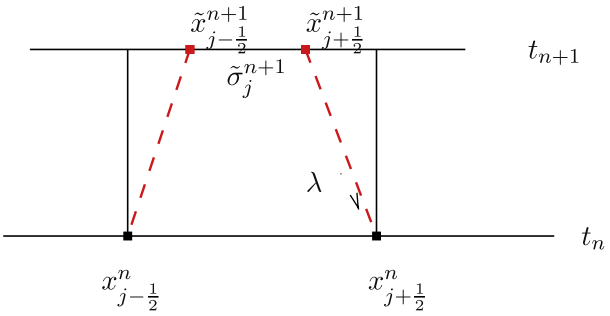


Fig. 3.1. 1D mesh redistribution from time t_n to t_{n+1}.

The formula of $\widetilde{U}^{n+1}_{j+\frac{1}{2}}$ is defined in this way

$$\widetilde{U}^{n+1}_{j+\frac{1}{2}} := R^A(\lambda; U_L, U_R) + \Delta t_n \left(\frac{\partial U}{\partial t} + \lambda \frac{\partial U}{\partial x}\right)^n_{j+\frac{1}{2}}. \tag{3.13}$$

This is the Taylor expansion for $U\left(x_{j+\frac{1}{2}}, t_{n+1}\right)$ along the direction λ, $R^A(\lambda; U_L, U_R)$ is the Riemann solution for the 1D Euler equations along the direction λ and the directional derivatives $\frac{\partial U}{\partial t} + \lambda \frac{\partial U}{\partial x}$ can be calculated within the framework of paper [5]. We write out the derivation procedure and often used coefficients in Appendix A.

Another natural way to define this new primitive slope $\tilde{\sigma}^{n+1}_j$ is to interpolate it from the old slope $\sigma^{[\nu]}_j$ after each iteration step. For simplicity, in this paper we use a conservative interpolation

$$\sigma^{[\nu+1]}_j = \frac{\Delta\tilde{x}^{[\nu]}_j \sigma^{[\nu]}_j - \left((c\sigma)^{[\nu]}_{j+\frac{1}{2}} - (c\sigma)^{[\nu]}_{j-\frac{1}{2}}\right)}{\Delta\tilde{x}^{[\nu]}_j}, \tag{3.14}$$

where $(c\sigma)^{[\nu]}_{j+\frac{1}{2}}$ is the value of $(c\sigma)^{[\nu]}$ on the interface $x_{j+\frac{1}{2}}$. At last, let $\tilde{\sigma}^{n+1}_j := \sigma^{[\mu]}_j$, where $[\mu]$ is the total number of iteration step.

For 1D examples, both methods work almost the same. However, it is quite hard to extend the first method to 2D cases since the cell interfaces might be curved and it is difficult to define the generalized Riemann problem. So we are forced to use the conservative interpolation, and extend (3.14) as

$$\left|\widetilde{A}_{ij}\right|\left(\tilde{\sigma}_z{}^{[\nu+1]}\right)_{ij} = |A_{ij}|\left(\sigma^{[\nu]}_z\right)_{ij} - \left[(n_c\sigma_z)^{[\nu]}_{l_2} + (n_c\sigma_z)^{[\nu]}_{l_4}\right] - \left[(n_c\sigma_z)^{[\nu]}_{l_1} + (n_c\sigma_z)^{[\nu]}_{l_3}\right], \tag{3.15}$$

where $(n_c\sigma_z)^{[\nu]}_{l_k}$ denote values of $(n_c\sigma_z)^{[\nu]}$ on the associated boundary l_k, z is x or y, $n_c = c^x\mu + c^y\nu$, and $(c^x, c^y) = (x - \tilde{x}, y - \tilde{y})$.

Remark 1. It is not necessary to remap the primitive slope on the new mesh if the van Leer limiter is used. But for the GRP scheme this limiter smears slip lines and shocks too much compared with the minmod limiter in many cases. As in the numerical tests we will show in Figs. 5.4 and 5.15, the van Leer limiter does not work effectively. Fig. 5.4 shows the right slowly moving shock. When the adaptive GRP scheme is employed with the van Leer limiter, the small density oscillation is not smeared out completely. In Fig. 5.15 we show the spiral generated from the interaction of four vortex sheets (contact discontinuities). The spiral in Fig. 5.15 with the van Leer limiter is much rougher resolved compared with that using the minmod limiter in Fig. 5.14. Furthermore, we extract the data from the density profiles along the diagonal line $x = y$ in Figs. 5.14 and 5.15, respectively, and show them in Fig. 5.16. We observe that the minmod limiter is much more accurate than the van Leer limiter in this example.

3.4. The monitor function

The monitor function is one of the most important elements in the moving mesh algorithms. The appropriate choice of the monitor will produce grids with good quality in terms of smoothness, skewness, and aspect ratio. The conventional monitor functions usually depend on the magnitude of the gradient of the solutions, which may not be always effective to increase the grid concentration in the local regions containing shock waves and contact discontinuities or their interactions.

However, the choices of the different parameters in the monitor functions depend strongly on the computed problem, when we expect to present a "satisfactory" or "optimal" result. It is very important to further improve the present algorithm in order to reduce this dependence. A possible and good direction is to use the monitor function without artificial parameters given in the recent papers of Zegeling, de Boer, and Tang [39,10] and [16].

Since the choice of the monitor function is not the main task in our paper, so we will just use a traditional choice of an arclength-type monitor (AL-monitor), such as

$$\omega = \sqrt{1 + \alpha \nabla U \cdot \nabla U}, \tag{3.16}$$

where α is a nonnegative constant and problem dependent, and U may represent the physical variables, such as density, velocity, entropy, local Mach number and so on. As commonly done, some spatial smoothing procedure for the monitor function is used in order to avoid very singular mesh and/or large errors around the stiff solution area. The following are examples:

For the 1D case, we choose

$$\omega_j \leftarrow \frac{1}{2}\omega_j + \frac{1}{4}(\omega_{j-1} + \omega_{j+1}). \tag{3.17}$$

In the present paper the monitor function for the 1D Euler equations (2.1) is always chosen as

$$\omega = \sqrt{1 + \alpha_1 \left(\frac{u_\xi}{\max\limits_\xi |u_\xi|}\right)^2 + \alpha_2 \left(\frac{s_\xi}{\max\limits_\xi |s_\xi|}\right)^2}, \tag{3.18}$$

where entropy $s = p/\rho^\gamma$ and the parameters $\alpha_i, i = 1, 2$, are some nonnegative constants.

For the 2D case, we choose

$$\omega_{ij} \leftarrow \frac{1}{4}\omega_{ij} + \frac{1}{8}(\omega_{ij+1} + \omega_{ij-1} + \omega_{i+1j} + \omega_{i-1j}) + \frac{1}{16}(\omega_{i+1j+1} + \omega_{i-1j-1} + \omega_{i+1j-1} + \omega_{i-1j+1}). \tag{3.19}$$

It is found that the monitor function (3.18) enlarges the start-up error around a single stationary shock for the 1D case, as shown in Fig. 5.7, for which $\alpha_1 = 5, \alpha_2 = 1$. In this case, the monitor function with only velocity variable is an ideal candidate to overcome this error, as seen in Fig. 5.5, in which $\alpha_1 = 5, \alpha_2 = 0$. It is also observed that this velocity monitor function works efficiently for 2D Riemann problems, especially for the spiral of the interaction of the contact discontinuities. The Mach number monitor function is able to keep the straight shock, but the accuracy cannot be improved inside the contact discontinuity region. The entropy monitor function is also a good choice for the interaction problem of shocks but it will enlarge the start-up errors. Thus in our numerical tests, we can combine some of them to improve the numerical accuracy, or avoid using some of them to decrease the possible errors. The monitor functions for all 2D tests are outlined in Section 5.

4. The algorithm of adaptive GRP scheme

Our algorithm consists of two independent parts: PDE evolution and iterative mesh distribution. The first part is to evolve the compressible fluid flows with the GRP scheme. See Section 2. In each iteration step of the second part, mesh points are first redistributed by the Gauss–Seidel method, and then the conservative variables and primitive slopes are updated on the resulting new meshes by the conservative interpolation. The primitive slopes can be also directly computed from the generalized Riemann problem for 1D cases. See Section 3.3. We outline the 2D solution procedure in the following. The algorithm of 1D adaptive GRP scheme is similar.

Step 1. If $t_n = 0$, give an initial partition $\left\{x_{i+\frac{1}{2}j+\frac{1}{2}}, y_{i+\frac{1}{2}j+\frac{1}{2}}\right\}$ to the physical domain Ω_p and a uniform partition to the logical domain Ω_c. Compute the grid values U^0_{ij} and primitive slopes $(\sigma_x)^0_{ij}, (\sigma_y)^0_{ij}$ based on the initial data $U(x, y, 0)$. If $t_n > 0$, set $(x, y)^{[0]}_{ij} = (x, y)^n_{ij}, U^{[0]}_{ij} = U^n_{ij}, (\sigma^{[0]}_z)_{ij} = (\sigma_z)^n_{ij}, z = x$ or y.

Step 2. For $\nu = 0, 1, 2, \ldots \mu - 1$, redistribute the mesh as follows:

(a) Compute the new mesh points in accord with the Gauss–Seidel iteration (3.6).

(b) Update the solution variables $\left\{U^{[\nu+1]}_{ij}\right\}$ and the primitive slopes $(\sigma_z)^{[\nu+1]}_{ij}$ (z is x or y) on the new grid using the conservation interpolation (3.9) and (3.15).

(c) Repeat the procedure (a) and (b) for a fixed number μ or until $\sum_{ij}\left|(x, y)^{[\nu+1]}_{ij} - (x, y)^{[\nu]}_{ij}\right| \leqslant \varepsilon$.

Step 3. Set $A_{ij} = A^{[\nu+1]}_{ij}, U^n_{ij} = U^{[\nu+1]}_{ij}, (\sigma_z)^n_{ij} = (\sigma_z)^{[\nu+1]}_{ij}$ (z is x or y). Reconstruct the piecewise linear function $U_{A_{ij}}(x, y, t_n)$ according to (4.1). Compute U^{n+1}_{ij} and σ^{n+1}_{ij} from the procedure listed in Section 2.3 on the new mesh A_{ij}.

Step 4. Go to **Step 1.** if $t_{n+1} < T$.

Remark 2. In our 2D numerical experiments, the slopes $(\sigma_x)^{n+1}_{ij}$ and $(\sigma_y)^{n+1}_{ij}$ on the cell A_{ij} are evaluated from the following step:

(i) Calculate the primitive slopes $(\sigma_\xi)^{n+1}_{ij}$ and $(\sigma_\eta)^{n+1}_{ij}$ in the computational plane (ξ, η), which is similar to (2.18).

(ii) Compute the slopes $(\sigma_x)^{n+1}_{ij}$ and $(\sigma_y)^{n+1}_{ij}$,

$$\begin{aligned}(\sigma_x)^{n+1}_{ij} &= (\sigma_\xi)^{n+1}_{ij}(\xi_x)^{n+1}_{ij} + (\sigma_\eta)^{n+1}_{ij}(\eta_x)^{n+1}_{ij},\\ (\sigma_y)^{n+1}_{ij} &= (\sigma_\xi)^{n+1}_{ij}(\xi_y)^{n+1}_{ij} + (\sigma_\eta)^{n+1}_{ij}(\eta_y)^{n+1}_{ij},\end{aligned} \tag{4.1}$$

where

$$\begin{cases} \xi_x = \frac{y_\eta}{A}, \\ \xi_y = -\frac{x_\eta}{A}, \end{cases} \quad \begin{cases} \eta_x = -\frac{y_\xi}{A}, \\ \eta_y = \frac{x_\xi}{A}, \end{cases} \quad A = \begin{vmatrix} x_\xi & y_\xi \\ x_\eta & y_\eta \end{vmatrix}, \tag{4.2}$$

and z_ξ, z_η (z is x or y) are approximated by the central difference on every mesh A_{ij}.

5. Numerical examples

5.1. One-dimensional examples

We choose three well-understood one-dimensional examples to show the performance of the present adaptive GRP scheme for the 1D case. In all figures, the solid lines represent the exact solutions, while the red dotted lines stand for the numerical solutions The parameter α in the minmod limiter (2.6) is $\alpha = 1.9$ and the polytropic index γ is taken as 1.4 for air unless explicitly stated otherwise. The vertical velocity $v \equiv 0$ is set for (2.1).

Example 5.1 (*Sod problem*). Our first example is the Sod shock tube problem [29]. The initial data are

$$(\rho, u, p)(x, 0) = \begin{cases} (1, 0, 1), & 0 < x < 0.5, \\ (0.125, 0, 0.1), & 0.5 < x < 1. \end{cases} \tag{5.1}$$

The exact solution consists of a leftward moving rarefaction wave, a contact discontinuity, and a rightward moving shock wave. Fig. 5.1 is displayed for the numerical result by the adaptive scheme at time $t = 0.15$ with 60 grid points, for which the CFL number is chosen to be 0.9, and the monitor function is given by the formula (3.18) with $\alpha_1 = 1.0, \alpha_2 = 1.0$. We can see that the adaptive GRP scheme does very well in capturing these three elementary waves.

Example 5.2 (*Nearly stationary strong shock*). Initially, we have

$$(\rho, u, p)(x, 0) = \begin{cases} (4.0, -0.3, 4.0/3.0), & 0 \leqslant x < 0.2, \\ (1.0, -1.3, 10^{-6}), & 0.2 < x \leqslant 1. \end{cases} \tag{5.2}$$

The polytropic index is taken to be $\gamma = 5/3$ for the helium. This example involves a very strong nearly stationary shock, whose exact speed is 3.4052×10^{-2}. This is an almost infinite shock in the sense that the density ratio is close to its maximum. The "wavelike" behavior will be generated by many high resolution numerical schemes, as pointed out [1] and shown

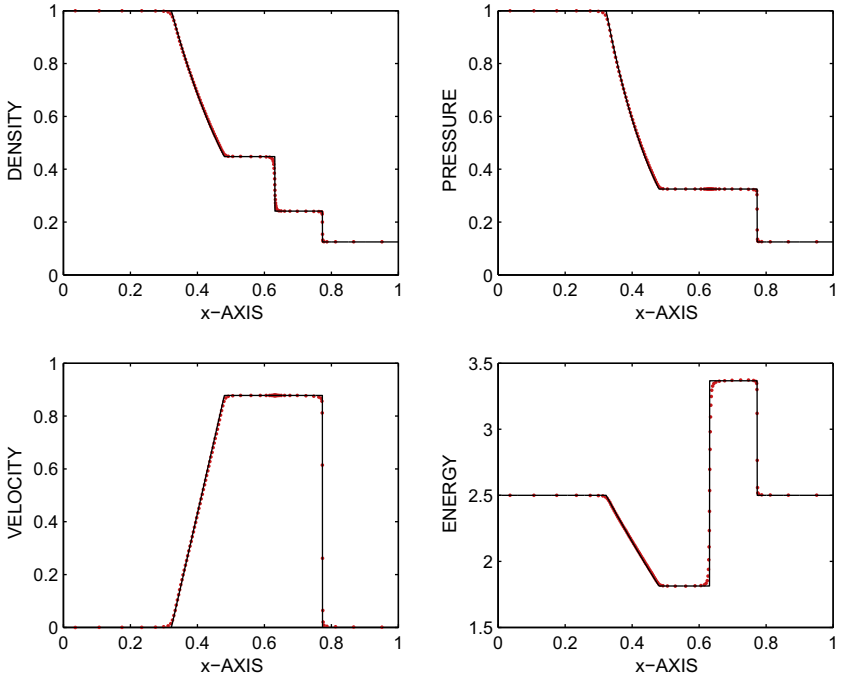


Fig. 5.1. The Sod problem by the adaptive GRP scheme: 60 grids points are used.

in Fig. 5.2, which is obtained using the GRP scheme [5] with 100 mesh points. This oscillation can be smoothed out by introducing the adaptive mesh redistribution mechanism. In Fig. 5.3, we use the present adaptive GRP scheme (60 grid points) with the minmod limiter (2.6), while in Fig. 5.4 we use the same grid points but with the van Leer limiter. The other parameters are the same: The monitor function is given by (3.18) with $\alpha_1 = 1, \alpha_2 = 1$; the CFL number is 0.9 and the output time is $t = 20$. Both results are almost consistent and satisfactory. However, there is still a small oscillation around the shock front of

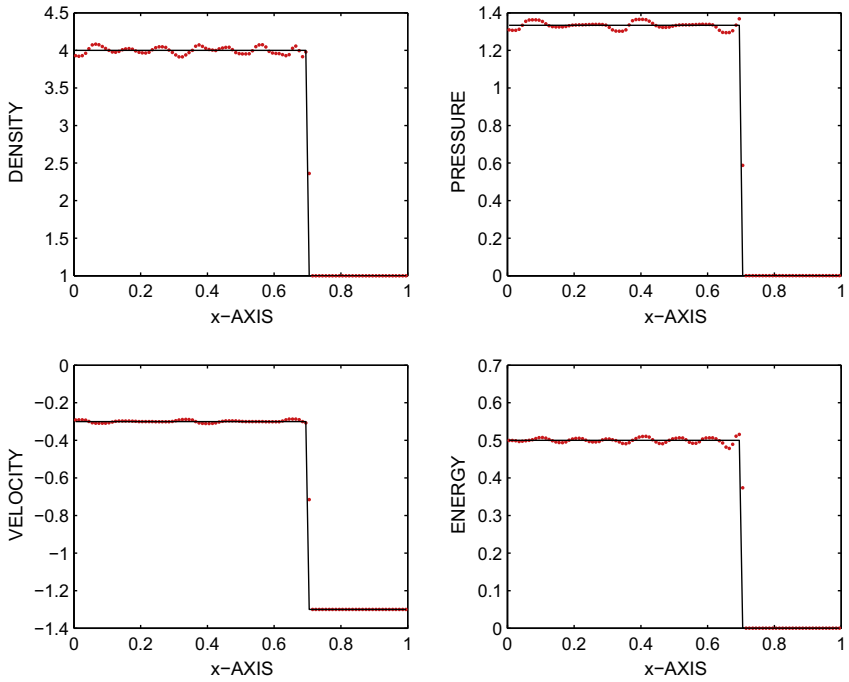


Fig. 5.2. Almost stationary shock by the direct Eulerian GRP scheme with 100 grid points at time $t = 20$.

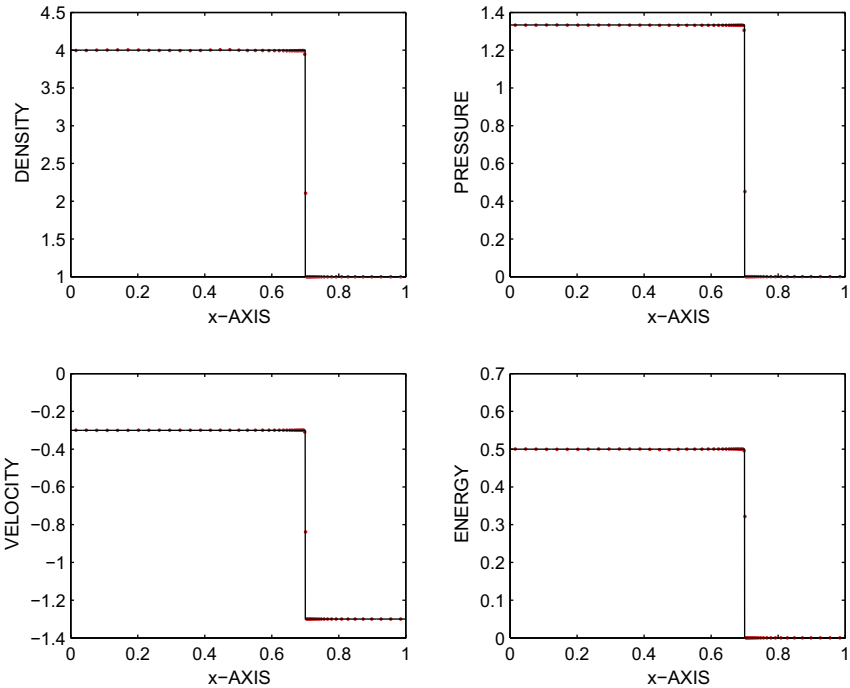


Fig. 5.3. Almost stationary shock by the adaptive GRP scheme with the minmod limiter: 60 grid points are used at time $t = 20$.

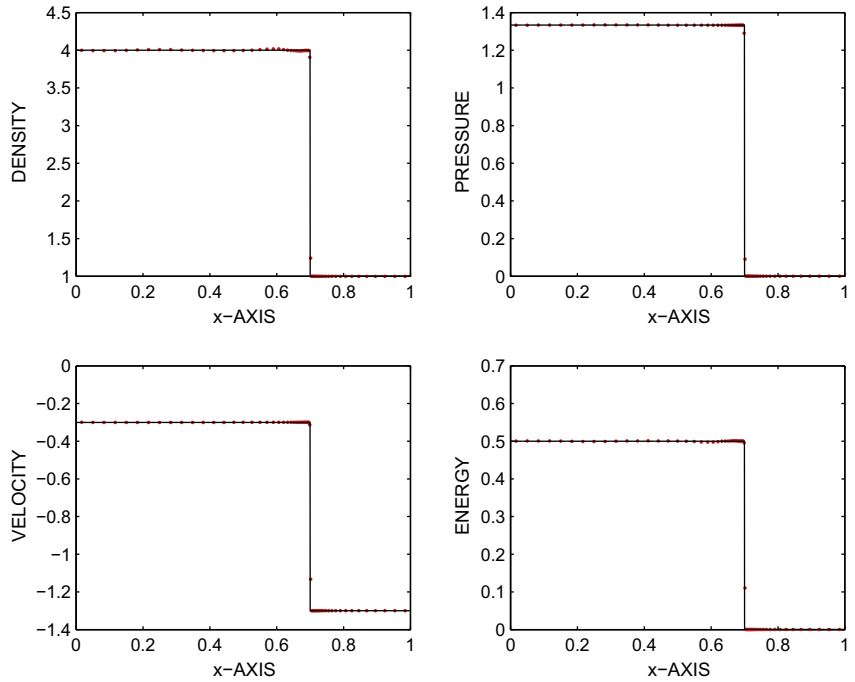


Fig. 5.4. The stationary shock by the adaptive GRP scheme with the van Leer limiter: 60 grid points are used at time $t = 20$.

the density plot in Fig. 5.4. Just as we have already pointed out, the minmod limiter is a little bit better than the van Leer limiter in our adaptive GRP scheme.

Example 5.3 (*Start-up errors in a single shock*). It is observed that there is a start-up error around the initial discontinuity point for the single shock wave, and it is still unclear about the mechanism of this phenomenon. In two-dimensional Riemann problem this phenomenon is quite common, refer to Fig. 3.3 in [19] and Fig. 11 in [26]. We take a look at the following example of a single shock. The initial data is taken as

$$(\rho, u, p)(x, 0) = \begin{cases} (0.532258, 1.206045, 0.3), & 0 \leqslant x < 0.5, \\ (1.5, 0.0, 1.5), & 0.5 < x \leqslant 1.0. \end{cases} \tag{5.3}$$

Fig. 5.5 shows the start-up error, which is obtained by the direct Eulerian GRP scheme [5] with 100 grid points. Obviously there is an error around the initial discontinuity point $x = 0.5$. Fig. 5.6 is the numerical result by the current adaptive GRP scheme with 60 grid points and the monitor function is (3.18) chosen with $\alpha_1 = 5$ and $\alpha_2 = 0$. The start-up error has been totally smeared out in Fig.5.6. However, we have to take caution in choosing the parameters for the monitor function (3.18). Fig. 5.7 is obtained by the adaptive GRP scheme with $\alpha_1 = 5$ and $\alpha_2 = 1$ for the monitor function (3.18). The start-up error has been enlarged significantly as both the density and entropy take effect in the monitor function (3.18). This shows that this start-up error may be caused by the violent variation of the entropy. The output time of the all numerical results is at $t = 0.15$ and the CFL number is 0.9.

5.2. Two-dimensional examples

We choose several two-dimensional Riemann problems and the well-known double Mach reflection problem to demonstrate the capabilities of the present adaptive GRP scheme for 2D cases. The two-dimensional Riemann problems were originally proposed by Zhang and Zheng [41], and then followed by many numerical simulations [28,26,19]. Systematic treatments can be found in [23,40]. The flow patterns are quite complex, including the Mach reflection, rolling up of slip lines, formation of shocks and much more. Nowadays the two-dimensional Riemann problems have been widely used for checking the resolution as well as efficiency of numerical schemes. We present the density contour curves and the corresponding mesh distribution for tested problems. The initial data for each example consist of a constant state in each quadrant. Furthermore, the initial data are designed so that only one elementary wave (a shock, a rarefaction wave or a contact discontinuity) emanates from each initial discontinuity along the coordinate axes. The notations (ρ_i, u_i, v_i, p_i) are used to express the constant state in the i-th quadrant, $i = 1, 2, 3, 4$; and other conventions are the same as in [23]. The parameter α in the minmod limier (2.6) is 1.9; the polytropic index γ is taken as 1.4 and the CFL number is taken 0.5 unless explicitly stated

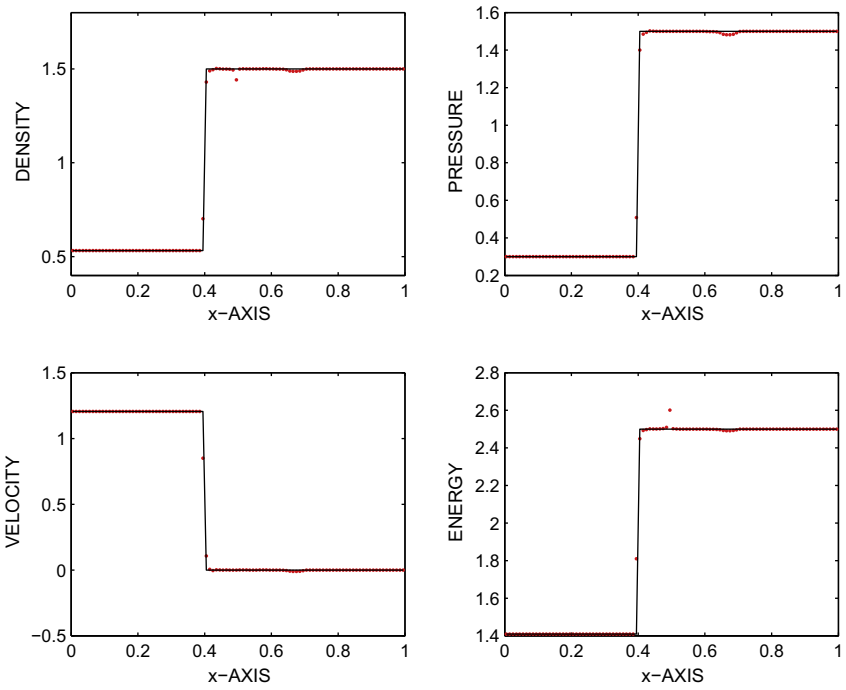


Fig. 5.5. A single shock by the direct Eulerian GRP scheme: 100 grid points are used.

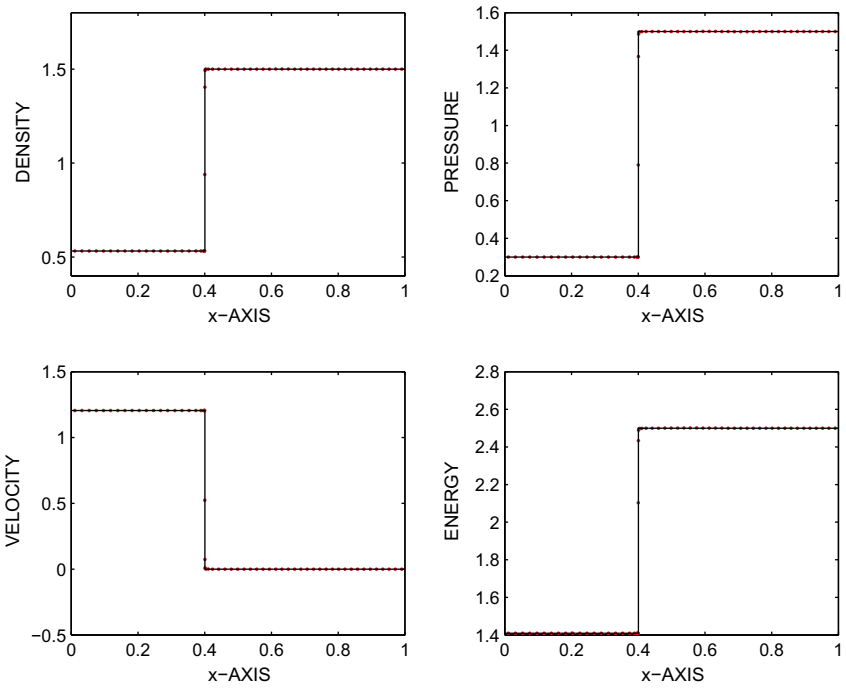


Fig. 5.6. A single shock by the adaptive GRP scheme: 60 grid points are used.

otherwise. The double Mach reflection problem can be found in many references and regarded locally as a two-dimensional Riemann problem. All the experiments have been performed on a workstation with Intel(R) Core(TM) 2 Quad CPU Q660 @2.40 GHZ.

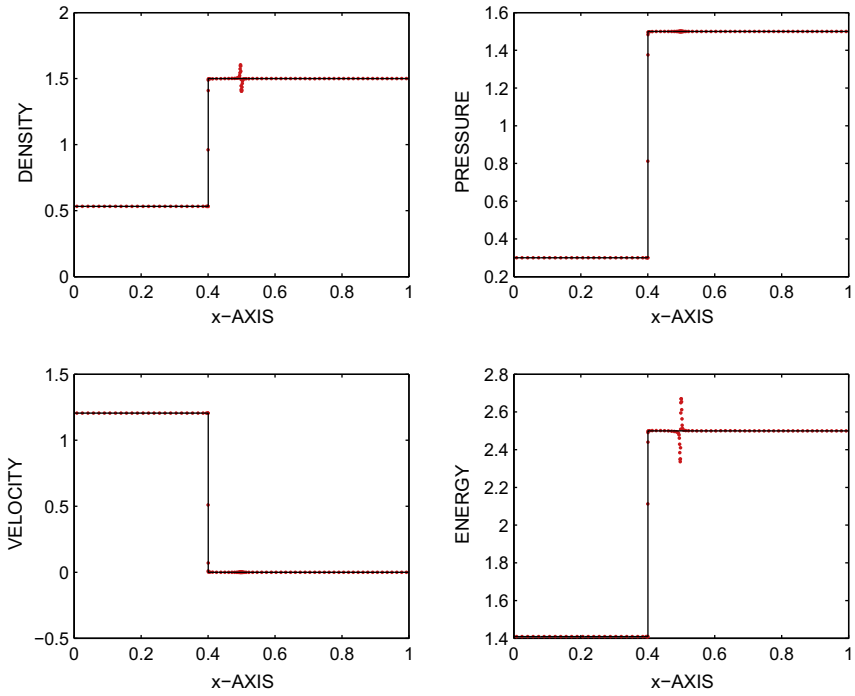


Fig. 5.7. A single shock by the adaptive GRP scheme with the entropy monitor function: 100 grid points are used.

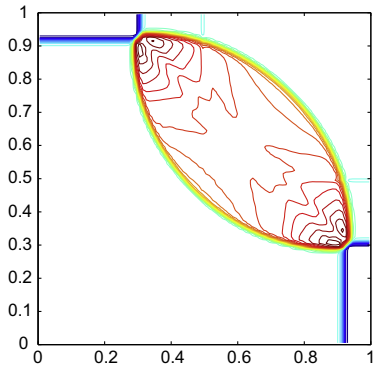


Fig. 5.8. The GRP scheme combined the Strang splitting method with 100×100 meshes. The CPU time is 23.218 s.

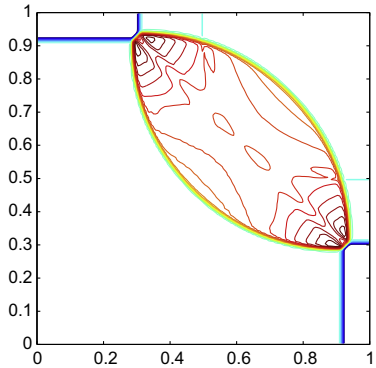


Fig. 5.9. The GRP scheme combined the Strang splitting method with 200×200 meshes. The CPU time is 194.390 s.

Example 5.4 (*Interaction of four shocks* $(S_{12}^- S_{23}^+ S_{34}^- S_{41}^+)$). The initial data are chosen to be

$$(\rho,u,v,p)(x,y,0)=\begin{cases}(1.1,0.0,0.0,1.1), & x>0.5,\ y>0.5,\\ (0.5065,0.8939,0.0,0.35), & x<0.5,\ y>0.5,\\ (1.1,0.8939,0.8939,1.1), & x<0.5,\ y<0.5,\\ (0.5065,0.0,0.8939,0.35), & x>0.5,\ y<0.5.\end{cases} \tag{5.4}$$

Initially a single planar shock emanates from each coordinate axis. The four shocks interact as time evolves, and a very complicated wave pattern emerges. It includes triple points, Mach stems and contact discontinuities, etc. If we regard the symmetric axis $x=y$ as the rigid wall, then the one-fourth (northeastern) part is just the Mach reflection case. This example is just the same as Fig. 4 in [26] and Fig. 3.4 in [19].

Figs. 5.8, 5.9, 5.10and 5.11 display the contours of the density at $t=0.25$ by the GRP scheme combined the Strang splitting method and the adaptive GRP scheme separately. The results indicate that the shocks and contact discontinuities are much better resolved by the adaptive GRP method on 50×50 mesh, compared to those on 100×100 uniform mesh, although the CPU time is spent nearly 68% more. However, compared to those on 200×200 uniform mesh, the current adaptive GRP method (on 50×50 mesh) uses less CPU time but obtain almost the same result. Here we also provide the wave configuration on 100×100 mesh by the adaptive GRP scheme, see Fig. 5.11. It is resolved with almost the same accuracy as [19,26] on 400×400 uniform mesh (see Figs. 5.12 and 5.13).

Example 5.5 (*Interaction of vortex sheets and formation of spirals* $(J_{12}^- J_{23}^- J_{34}^- J_{41}^-)$). The Riemann initial data are chosen to be

$$(\rho,u,v,p)=\begin{cases}(0.1,0.5,-0.5,10), & x>0.5,\ y>0.5,\\ (0.15,0.5,0.5,10), & x<0.5,\ y>0.5,\\ (0.09,-0.5,0.5,10), & x<0.5,\ y<0.5,\\ (0.05,-0.5,-0.5,10), & x>0.5,\ y<0.5.\end{cases} \tag{5.5}$$

This example typically describes the interaction of vortex sheets in 2D compressible fluid flows. Initially four vortex sheets emanates from the initial discontinuities. Since the slip lines have the same sign, the structure of the solution is a spiral turning clockwise in subsonic domain and the slip lines (vortex sheets) roll up around its center. Here we choose such initial data,

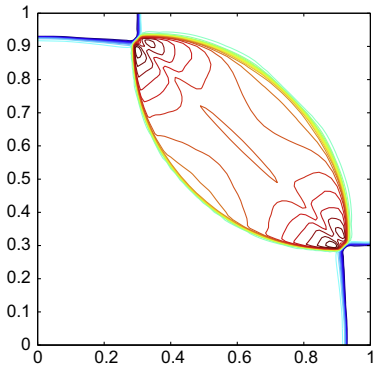


Fig. 5.10. The adaptive GRP scheme with $\omega=\sqrt{1+5.0\|\nabla_{\xi\eta}s\|_{L^2}^2}$ and 50×50 meshes. The CPU time is 72.484 s.

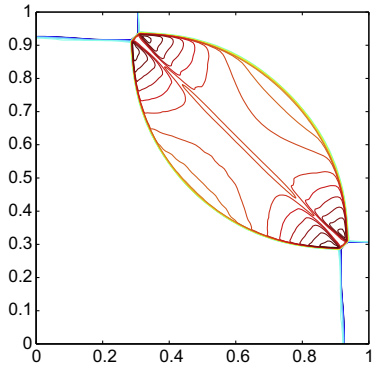
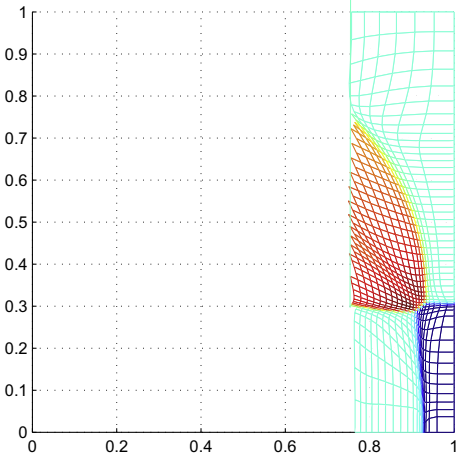


Fig. 5.11. The adaptive GRP scheme with $\omega=\sqrt{1+2.0\|\nabla_{\xi\eta}s\|_{L^2}^2}$ and 100×100 meshes. The CPU time is 210.906 s.



for which a spiral is generated at the center. Theoretically, the spiral should revolve infinitely [23]. Thus the more times the spiral revolves numerically, the better resolution the scheme has. Fig. 5.14 is the numerical profile of the density at time

$t = 0.2$, which is obtained with 100×100 meshes and the minmod limiter (2.6). Fig. 5.15 is almost the same as Fig. 5.14 except that the van Leer limiter is used. The monitor function for Figs. 5.15 and 5.14 is $\omega = \sqrt{1 + 1.0\left(u_\xi^2 + u_\eta^2 + v_\xi^2 + v_\eta^2\right) + 3.0\left(\rho_\xi^2 + \rho_\eta^2\right)}$. We can see that the adaptive GRP scheme works well on this example. Fig. 5.16 shows the density profiles along $x - y = 0$, by taking data from Figs. 5.14 and 5.15. Oscillations in terms of the density show how the vortex sheets revolve around the center.

Example 5.6 (*Double Mach reflection*). This problem was extensively studied by Woodward and Collela in [38] and later by many others. We use exactly the same setup as in [31], i.e. the same initial boundary conditions and the same solution domain $\Omega_p = [0, 4] \times [0, 1]$. Initially a right moving Mach 10 shock is positioned at $x = \frac{1}{6}, y = 0$ and makes a 60° angle with the x-axis. More precisely, the initial data are

$$U = \begin{cases} U_L, & y \geqslant h(x, 0), \\ U_R, & \text{otherwise}, \end{cases} \tag{3.6}$$

where the left and right states as well as the position of the shock wave are, respectively,

$$\begin{cases} U_L = (8, 57.1579, -33.0012, 563.544)^\top, \\ U_R = (1.4, 0.0, 0.0, 2.5)^\top, \\ h(x, t) = \sqrt{3}(x - \frac{1}{6}) - 20t. \end{cases} \tag{3.7}$$

The numerical results at $t = 0.2$ is in Fig. 5.17, which is obtained with 320×80 cells. The CFL number in this computation is 0.5 and the monitor function is $\sqrt{1 + 0.15\left(\rho_\xi^2 + \rho_\eta^2\right)}$. Fig. 5.18 is the detail of the double mach reflection region near the

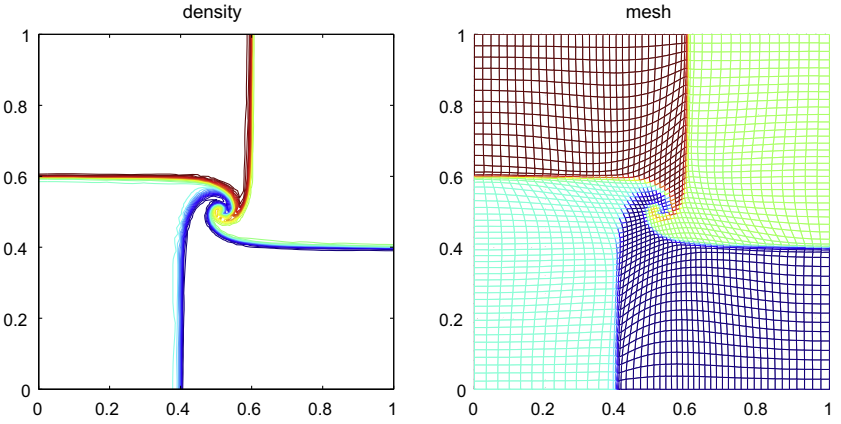


Fig. 5.15. The same as Fig. 5.14 but with the van Leer limiter.

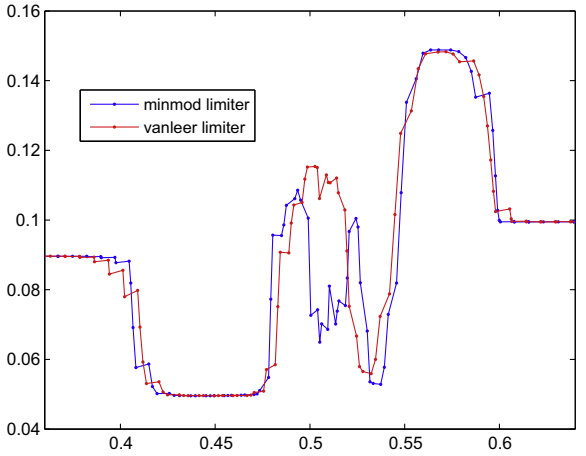


Fig. 5.16. A comparison of the density along the diagonal line $x - y = 0$ of the spiral with the minmod limiter and the van Leer limiter, respectively.

double-Mach stem with the monitor function $\sqrt{1+0.015\left(\rho_\xi^2+\rho_\eta^2\right)}$. This result is well consistent with those by many well-known schemes.

6. Conclusions and remarks

In this paper an adaptive GRP scheme for two-dimensional compressible fluid flows is developed, by combining the direct Eulerian GRP scheme [5] and the adaptive moving mesh method [31] in order to improve the accuracy of the numerical solution with fewer points and effectively reduce possible errors (oscillations), which may be produced by many Godunov-type schemes (including the GRP scheme). Then several one- and two-dimensional examples are provided to demonstrate the capabilities of this scheme. We summarize the features of the adaptive GRP as follows:

(i) Both the GRP scheme and the adaptive method are very flexible. When they are combined together, many numerical deficiencies can be offset. Of course we can also use other singularity tracking methods, such as that proposed in [2,15]. However the present scheme is much more easily coded and applicable to more complex geometrical domains because the underlying GRP scheme is devised on arbitrary quadrangular meshes.

(ii) The present adaptive scheme can also be devised based on other (triangular) meshes and extended to three-dimensional cases.
(iii) The CPU time is competitive when compared to the original 2D GRP scheme with the Strang splitting method.
(iv) From the numerical examples, this scheme at least has the same resolution in capturing hydrodynamic singularities (shocks or contact discontinuities).

Acknowledgment

Jiequan Li is partially supported by the Key Program from Beijing Educational Commission (No. KZ200510028018), 973 Project (No. 2006CB805902), PHR (IHLB) and NSFC (No. 10971142). Huazhong Tang was partially supported by the National Basic Research Program under the Grant 2005CB321703, the National Natural Science Foundation of China (Nos. 10925101 and 10828101), Doctoral Program of Education Ministry of China (No. 20070001036), and Program for New Century Excellent Talents in University (NCET-07-0022).

Appendix A. The solution procedure for the generalized Riemann problem along λ direction

Assume $U(x,t)$ is the solution of the generalized Riemann problem (2.1) and (2.7) with $v=0$. In the framework of paper [5], we approximate $U(x,t)$ along λ with the formula

$$U(x,t) := R^A(\lambda; U_L, U_R) + t\lim_{s\to 0}\left(\frac{\partial U}{\partial t} + \lambda\frac{\partial U}{\partial x}\right)(\lambda, s). \tag{6.1}$$

We have the linear algebraic equations

$$\begin{cases} a_L\left(\frac{Dp}{Dt}\right)_* + b_L\left(\frac{Du}{Dt}\right)_* = d_L, \\ a_R\left(\frac{Dp}{Dt}\right)_* + b_R\left(\frac{Du}{Dt}\right)_* = d_R. \end{cases} \tag{6.2}$$

All of coefficients are defined in paper [5]. Then we have

$$\begin{aligned} \left(\frac{\partial u}{\partial t}\right)_* &= \left(\frac{Du}{Dt}\right)_* + \frac{u_*}{\rho_* c_*^2}\left(\frac{Dp}{Dt}\right)_*, \\ \left(\frac{\partial p}{\partial t}\right)_* &= \left(\frac{Dp}{Dt}\right)_* + \rho_* u_*\left(\frac{Du}{Dt}\right)_*. \end{aligned} \tag{6.3}$$

Let $\frac{\widetilde{D}}{Dt} = \frac{\partial}{\partial t} + \lambda\frac{\partial}{\partial x}$ and $U^* = U(\rho^*, u^*, p^*)$ be the state that the line λ is located. If U^* is not included in the rarefaction waves, we obtain

$$\begin{aligned} \left(\frac{\widetilde{D}p}{\widetilde{D}t}\right)_* &= \frac{u_*^2 - c_*^2 - \lambda u_*}{u_*^2 - c_*^2}\left(\frac{\partial p}{\partial t}\right)_* + \frac{\lambda\rho_* c_*^2}{u_*^2 - c_*^2}\left(\frac{\partial u}{\partial t}\right)_*, \\ \left(\frac{\widetilde{D}u}{\widetilde{D}t}\right)_* &= \frac{u_*^2 - c_*^2 - \lambda u_*}{u_*^2 - c_*^2}\left(\frac{\partial u}{\partial t}\right)_* - \frac{\lambda}{\rho_*(u_*^2 - c_*^2)}\left(\frac{\partial p}{\partial t}\right)_*. \end{aligned} \tag{6.4}$$

If U^* is in the rarefaction wave, we have

$$\left(\frac{\widetilde{D}u}{\widetilde{D}t}\right)_* = d_K(0), \quad \left(\frac{\widetilde{D}p}{\widetilde{D}t}\right)_* = \rho_* u_* d_K(0), \tag{6.5}$$

where $K = L$ or R, $d_K(0)$ is defined in paper [5]. The limiting value $\left(\frac{\widetilde{D}\rho}{\widetilde{D}t}\right)_*$ depends on wether the contact discontinuity propagates to the left or right part of line λ. and it is calculated in accord with the wave configuration. The results are listed as follows.

(i) If λ is located in the region of the state which connects to the initial state U_K ($K = L$ or R) with a rarefaction wave,

$$\left(\frac{\widetilde{D}\rho}{\widetilde{D}t}\right)_* = \frac{1}{c_*^2}\left(\left(\frac{\widetilde{D}p}{\widetilde{D}t}\right)_* + (\gamma-1)\rho_*(u_* - \lambda)\left(\frac{c_*}{c_K}\right)^{(1+\mu^2)/\mu^2} T_K S_K'\right), \tag{6.6}$$

where $T_K S_K'$ is given in paper [5].

(ii) If λ is located in the region of the state which connects to the initial state U_K ($K = L$ or R) with a shock wave,

$$\left(\frac{\widetilde{D}\rho}{\widetilde{D}t}\right)_* = \frac{(\sigma-\lambda)\left(\frac{\partial\rho}{\partial t}\right)_* + \lambda\left(H_1\left(\left(\frac{\widetilde{D}p}{\widetilde{D}t}\right)_* - \rho_*(\sigma - u_*)\left(\frac{\widetilde{D}u}{\widetilde{D}t}\right)_*\right) + Q\right)}{\sigma} \tag{6.7}$$

where $Q = (\sigma - u_K)H_2 p'_K + (\sigma - u_K)H_3 \rho'_K - \rho_K(H_2 c_K^2 + H_3)u'_K$ and σ is the shock speed. $(\frac{\partial \rho}{\partial t})_*, \sigma$ and $H_k, (k = 1, 2, 3)$ are defined in paper [5].

References

[1] M. Ben-Artzi, J. Falcovitz, A second-order Godunov-type scheme for compressible fluid dynamics, J. Comput. Phys. 55 (1984) 1–32.
[2] M. Ben-Artzi, J. Falcovitz, An upwind second-order scheme for compressible duct flows, SIAM J. Sci. Stat. Comput. 7 (1986) 744–768.
[3] M. Ben-Artzi, J. Falcovitz, Generalized Riemann Problems in Computational Fluid Dynamics, Cambridge University Press, 2003.
[4] M. Ben-Artzi, J. Li, Hyperbolic balance laws: Riemann invariants and the generalized Riemann problem, Numer. Math. 106 (3) (2007) 369–425.
[5] M. Ben-Artzi, J. Li, G. Warnecke, A direct Eulerian GRP scheme for compressible fluid flows, J. Comput. Phys. 218 (2006) 19–34.
[6] J.U. Brackbill, An adaptive grid with directional control, J. Comput. Phys. 108 (1993) 38–50.
[7] J.U. Brackbill, J.S. Saltzman, Adaptive zoning for singular problems in two-dimensions, J. Comput. Phys. 46 (1982) 342–368.
[8] W. Cao, W. Huang, R.D. Russell, A study of monitor functions for two-dimensional adaptive mesh generation, SIAM J. Sci. Comput. 20 (1999) 1978–1999.
[9] W. Cao, W. Huang, R.D. Russell, An r-adaptive finite element method based upon moving mesh PDEs, J. Comput. Phys. 149 (1999) 221–244.
[10] A. van Dam, P.A. Zegeling, Balanced monitoring of flow phenomena in moving mesh methods, Commun. Comput. Phys. 7 (2010) 138–170.
[11] A. van Dam, P.A. Zegeling, A robust moving mesh finite volume method applied to 1D hyperbolic conservation laws from magnetohydrodynamics, J. Comput. Phys. 216 (2006) 526–546.
[12] S.F. Davis, 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.
[13] Y. Di, R. Li, T. Tang, P. Zhang, Moving mesh finite element methods for the incompressible Navier–Stokes equations, SIAM J. Sci. Comput. 26 (2005) 1036–1056.
[14] A.S. Dvinsky, Adaptive grid generation from harmonic maps on Riemannian manifolds, J. Comput. Phys. 95 (1991) 450–476.
[15] J. Falcovitz, A. Birman, A singularities tracking conservation laws scheme for compressible duct flows, J. Comput. Phys. 115 (1994) 431–439.
[16] W. Huang, Variational mesh adaptation: isotropy and equidistribution, J. Comput. Phys. 174 (2) (2001) 903–924.
[17] S.K. Godunov, A finite difference method for the numerical computation and discontinuous solutions of the equations of fluid dynamics, Mater. Sb. 47 (1959) 271–295.
[18] J. Han, H. Tang, An adaptive moving mesh method for multidimensional ideal magnetohydrodynamics, J. Comput. Phys. 220 (2007) 791–812.
[19] A. Kurganov, E. Tadmor, Solution of two-dimensional Riemann problems for gas dynamics without Riemann problem solvers, Numer. Methods Partial Diff. Eqs. 18 (2002) 584–608.
[20] J. Li, G. Chen, The generalized Riemann problem method for the shallow water equations with bottom topography, Int. J. Numer. Methods Eng. 65 (2006) 834–862.
[21] J. Li, T. Liu, Z. Sun, Implementation of the GRP scheme for computing radially symmetric compressible fluid flows, J. Comput. Phys. 228 (2009) 5867–5887.
[22] J. Li, Z. Sun, Remark on the generalized Riemann problem method for compressible fluid flows, J. Comput. Phys. 222 (2) (2007) 796–808.
[23] J. Li, T. Zhang, S. Yang, The Two-dimensional Riemann Problem in Gas Dynamics, Addison Wesley Longman, 1998.
[24] R. Li, T. Tang, P. Zhang, Moving mesh methods in multiple dimensions based on harmonic maps, J. Comput. Phys. 170 (2001) 562–588.
[25] R. Li, T. Tang, P. Zhang, A moving mesh finite element algorithm for singular problems in two and three space dimensions, J. Comput. Phys. 177 (2002) 365–393.
[26] X. Liu, P.D. Lax, Solution of two-dimensional Riemann problems of gas dynamics by positive schemes, SIAM J. Sci. Comput. 19 (1998) 319–340.
[27] K. Miller, R.N. Miller, Moving finite element. I, SIAM J. Numer. Anal. 18 (1981) 1019–1032.
[28] C.W. Schulz-Rinne, J.P. Collins, H.M. Glaz, Numerical solution of the Riemann problem for two-dimensional gas dynamics, SIAM J. Sci. Comput. 14 (1993) 1394–1414.
[29] G.A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, J. Comput. Phys. 27 (1978) 1–31.
[30] H. Tang, A moving mesh method for the Euler flow calculations using a directional monitor function, Commun. Comput. Phys. 1 (2006) 656–676.
[31] H. Tang, T. Tang, Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws, SIAM J. Numer. Anal. 41 (2003) 487–515.
[32] T. Tang, Moving mesh methods for computational fluid dynamics, Contem. Math. 383 (2005) 185–218.
[33] E.F. Toro, Riemann solvers and numerical methods for fluid dynamics: A practical introduction, Springer, 1997.
[34] B. van Leer, Towards the ultimate conservative difference scheme V, J. Comput. Phys. 32 (1979) 101–136.
[35] C. Wang, H. Tang, T. Liu, An adaptive ghost fluid finite volume method for compressible gas–water simulations, J. Comput. Phys. 227 (2008) 6385–6409.
[36] D. Wang, X. Wang, A three-dimensional adaptive method based on the iterative grid redistribution, J. Comput. Phys. 199 (2004) 423–436.
[37] A. Winslow, Numerical solution of the quasi-linear Poisson equation, J. Comput. Phys. 1 (1967) 149–172.
[38] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1984) 115–173.
[39] P.A. Zegeling, W.D. de Boer, H.Z. Tang, Robust and efficient adaptive moving mesh solution of 2D Euler equation, Contem. Math. 383 (2005) 375–386.
[40] Y. Zheng, Systems of conservation laws: two-dimensional Riemann problems, Birkhäuser, 2001.
[41] T. Zhang, Y. Zheng, Conjecture on the structure of solutions of the Riemann problem for two-dimensional gas dynamics systems, SIAM J. Math. Anal. 21 (1990) 593–630.