Journal of Computational Physics 242 (2013) 367–386

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

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

The adaptive GRP scheme for compressible fluid flows over unstructured meshes

Jiequan Li [a,*], Yongjin Zhang [b]

[a] School of Mathematical Sciences, Beijing Normal University, 100875 Beijing, PR China
[b] School of Mathematics and Information Sciences, Henan Polytechnic University, 454000 Jiaozuo, PR China

ARTICLE INFO

Article history:
Received 18 July 2011
Received in revised form 21 January 2013
Accepted 2 February 2013
Available online 22 February 2013

Keywords:
GRP solver
Adaptive moving mesh method
Unstructured triangular meshes
Conservative interpolation
Least squares method

ABSTRACT

Unstructured mesh methods have attracted much attention in CFD community due to the flexibility for dealing with complex geometries and the ability to easily incorporate adaptive (moving) mesh strategies. When the finite volume framework is applied, a reliable solver is crucial for the construction of numerical fluxes, for which the generalized Riemann problem (GRP) scheme undertakes such a task in the sense of second order accuracy. Combining these techniques yields a second order accurate adaptive generalized Riemann problem (AGRP) scheme for two dimensional compressible fluid flows over unstructured triangular meshes. Besides the generation of meshes, the main process of this combination consists of two ingredients: Fluid dynamical evolution and mesh redistribution. The fluid dynamical evolution ingredient serves to evolve the compressible fluid flows on a fixed nonuniform triangular mesh with the direct Eulerian GRP solver. The role of the mesh redistribution is to redistribute mesh points on which a conservative interpolation formula is adopted to calculate the cell-averages for the conservative variables, and the gradients of primitive variables are reconstructed using the least squares method. Several examples are taken from various contexts to demonstrate the performance of such a program.

1. Introduction

Due to complexity geometries of computational domains and the advent of adaptive mesh refinement technology, unstructured mesh methods become necessary and preferable in practical applications in the past decades [36]. The feature of being easily incorporated into the finite volume framework makes it possible to combine them with high resolution algorithms for the computation of compressible fluid flows in the community of CFD. In this context, the request of efficient and reliable solvers becomes crucial in the construction of numerical fluxes, for which there are many successful exact or approximate Riemann solvers [42]. As a second order extension of the Godunov method [18], the GRP solver, the name after the high resolution of the associated generalized Riemann problem (GRP) method [2], provides a high order strategy (solver) to construct numerical fluxes and its excellent performance has been proven in various problems [3–5,7,28,48,49].

This paper aims to combine these techniques together to develop a high resolution adaptive method on unstructured triangular meshes based on the GRP solver to form the adaptive generalized Riemann problem solver (AGRP). The process consists of two parts (besides the generation of meshes): The fluid dynamical evolution and the adaptive moving mesh redistribution. In the first part we adopt the GRP solver to construct the numerical flux on each cell interface in order to achieve the dynamical evolution of compressible fluid flows in the finite volume framework. The GRP solver is based on

* Corresponding author.
E-mail addresses: jiequan@bnu.edu.cn (J. Li), zhyj@hpu.edu.cn (Y. Zhang).

0021-9991/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2013.02.003

the analytical resolution of the associated generalized Riemann problem [2,5]. The original version uses the Lagrangian framework taking the advantage that contact discontinuities are fixed to coincide with cell interfaces and then passes to the Eulerian framework [2]. In order to avoid the delicate passage from the Lagrangian to the Eulerian, a direct Eulerian GRP scheme was developed using the fine properties of Riemann invariants [7]. In fact, it has been proven that the direct Eulerian GRP has more potential in a broad spectrum in the sense, for example, that it can be extended to one- or multi–dimensional systems of general hyperbolic balance laws [6] and be easily incorporated with the adaptive moving mesh technology [20]. Recent comparison with other schemes [29] also shows the competitiveness of the GRP solver. However, these contributions are almost all made on structured quadrangular meshes except a recent contribution based on the Lagrangian GRP solver for two–dimensional compressible fluid flows on unstructured meshes by Maire [34]. Hence it is worthwhile to extend the direct Eulerian GRP solver on unstructured meshes, including triangular meshes. It turns out that the current AGRP solver performs well for most cases, but cannot surmount the common defects caused by the unstructured meshes such as the distortion of symmetry.

The adaptive moving mesh method adopted here was proposed in [41]. The mesh redistribution step serves to locally cluster or spread out mesh points according to the solution variations so that singular or nearly singular solutions in fairly localized regions can be well resolved dynamically, such as shock waves and vortices. Nevertheless, this adaptive redistribution may greatly distort the shape of meshes even though they are regular or structured before the redistribution. At this point the unstructured mesh methods exhibit more advantages. In the past several decades, there were indeed a lot of important progress in moving mesh methods for partial differential equations, including the variational approach of Winslow [45], Brackbill et al. [8,9], and Li et al. [16,31,32]; moving finite element methods of Miller et al. [37] and Davis et al. [15]; and moving mesh PDEs of Cao et al. [11,12]. Some recent works on the moving mesh methods can be found in [14,22,44,47] and references therein. The adaptive moving mesh method was developed on unstructured meshes in [13] even for multi–medium computation. The application of the GRP solver on the adaptive structured quadrangular meshes can be found in [20]. This paper extends [13,20] to apply the GRP solver on the adaptive moving triangular meshes.

As the unstructured mesh technology is applied, the data reconstruction raises a new issue, particularly for high resolution algorithms. In this paper, the least squares method is applied to reconstruct the gradient of flow variables. This method is independent of the mesh topology and only relies on a stencil that identifies relevant neighboring cells for use in terms of the gradient estimation. An intensive review on least squares methods can be found in [35] and references therein. Certainly, this reconstruction has to be modified with some limiter algorithms. In most cases the Barth limiter [1] is adopted except the steady flow simulation for which the Venkatakrishnan limiter [43] is used in order to accelerate the convergence rate.

This paper is organized as follows. In Section 2, the governing equations and some notations are given, and a direct Eulerian GRP scheme on unstructured triangular meshes for two dimensional compressible fluid flows is described. In Section 3, we present data reconstruction including some gradient reconstruction methods and limiters. In Section 4, the adaptive moving mesh method is illustrated. Section 5 shows the algorithm of the adaptive GRP scheme. Numerical experiments are carried out in Section 6 to demonstrate the accuracy, the efficiency and the performance of this AGRP scheme. A final conclusion is given in Section 7.

2. The GRP scheme on triangular meshes

In this section we will formulate the GRP method to solve two dimensional compressible fluid flows on fixed (but unstructured) triangular meshes and state the strategy how to construct numerical fluxes and achieve the GRP algorithm.

2.1. Governing equations

The governing equations for two dimensional unsteady inviscid compressible fluid flows over the physical domain Ω_p are described as,

$$U_t + \nabla \cdot \boldsymbol{F}(U) = 0, \quad \boldsymbol{x} \in \Omega_p,\ t > 0, \tag{2.1}$$

where $\boldsymbol{F}(U) = (f(U), g(U)), \nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})^\top$ and

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \quad f(U) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u(E+p) \end{pmatrix}, \quad g(U) = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}.$$

As usual, $\rho, p, \boldsymbol{u} = (u, v)^\top$ represent the density, the pressure and the velocity vector, respectively, the total energy is expressed as $E = \frac{1}{2}\rho|\boldsymbol{u}|^2 + \rho e, e$ is the internal energy. The fundamental thermodynamic relation

$$\mathrm{d}e = T\mathrm{d}S + \frac{p}{\rho^2}\mathrm{d}\rho$$

is used to close the system (2.1), where T and S are the absolute temperature and entropy, respectively. In our numerical examples, we usually use the state equation $e = \frac{p}{(\gamma-1)\rho}$, $\gamma > 1$ (the ratio of specific heats), for polytropic gases.

2.2. The GRP scheme on a triangular mesh

Given a (physical) domain Ω_p, we partition it as $\Omega_p = \bigcup_{i\in I} T_i$ using triangular meshes, where I is an index set, T_i is an arbitrary triangle with three vertices $\boldsymbol{x}_{ij}, j = 1, 2, 3$, and they are ordered in the counter-clockwise manner. Assume that a partition of the time interval $[0, T]$ is given as $\{t_{n+1} = t_n + \Delta t_n; t_0 = 0, \Delta t_n > 0, n \in \mathbb{N}\}$, where the time step Δt_n is determined by the stability condition in practice. Denote by $C_i = T_i \times [t_n, t_{n+1})$ a parallel polyhedra control volume with three lateral faces A_{ij}, and further by T_{ij} the j^{th} neighboring triangles to $T_i, j = 1, 2, 3$. The common boundary of A_{ij}, T_{ij} and T_i is ℓ_{ij}; $\bar{\boldsymbol{x}}_i$, $\bar{\boldsymbol{x}}_{ij}$ and $\hat{\boldsymbol{x}}_{ij}$ are the centroids of the triangles T_i, T_{ij} and the side ℓ_{ij}, respectively, as shown in Fig. 2.1.

Fix the control volume C_i. We formulate our scheme for (2.1) in the finite volume framework

$$U_i^{n+1} = U_i^n - \frac{1}{|T_i|}\sum_{j=1}^{3}\int_{A_{ij}} \boldsymbol{F}(U)\cdot \boldsymbol{n}_{ij} dS, \tag{2.2}$$

where $\boldsymbol{n}_{ij}$ is the unit outward normal vector of the boundary ℓ_{ij} pointing from T_i to T_{ij}, the quantity U_i^n represents the average of the solution vector $U(\boldsymbol{x}, t_n)$ over T_i at the time t_n,

$$U_i^n = \frac{1}{|T_i|}\int_{T_i} U(\boldsymbol{x}, t_n) d\boldsymbol{x},$$

and $|T_i|$ is the area of the triangle cell T_i,

$$|T_i| = \frac{1}{2}|\overrightarrow{\boldsymbol{x}_{i1}\boldsymbol{x}_{i2}} \times \overrightarrow{\boldsymbol{x}_{i1}\boldsymbol{x}_{i3}}|.$$

Then the GRP scheme is first to assume that the data at time $t = t_n$ are piecewise linear function

$$U(\boldsymbol{x}) = U_i^n + \boldsymbol{\sigma}_i^n \cdot (\boldsymbol{x} - \bar{\boldsymbol{x}}_i), \quad \boldsymbol{x} \in T_i, \tag{2.3}$$

and proceed to approximate the numerical fluxes through A_{ij} in (2.2) by using the mid–point rule

$$\int_{A_{ij}} \boldsymbol{F}(U)\cdot \boldsymbol{n}_{ij} dS \approx \boldsymbol{F}(U_{A_{ij}}^{n+\frac{1}{2}}) \cdot \boldsymbol{n}_{ij}|A_{ij}|. \tag{2.4}$$

In (2.3) $\boldsymbol{\sigma}_i^n$ is the gradient of U in the cell T_i at time t_n and it is to be constructed in Section 3, $|A_{ij}| = |\ell_{ij}|\Delta t_n$ is the area of A_{ij}, $|\ell_{ij}|$ is the length of the side ℓ_{ij}, and the centroid values $U_{A_{ij}}^{n+\frac{1}{2}}$ are computed from the following local one-dimensional generalized Riemann problem,

$$\begin{cases} \frac{\partial U}{\partial t} + \frac{\partial H(U;\boldsymbol{n}_{ij})}{\partial \xi} = 0, \\ U(\zeta, 0) = \begin{cases} U_L + \xi U'_{\xi L}, & \xi < 0, \\ U_R + \xi U'_{\xi R}, & \xi > 0, \end{cases} \end{cases} \tag{2.5}$$

where the notations $U_L, U_R, U'_{\xi L}$ and $U'_{\xi R}$ are

$$\begin{cases} U_L = U_{T_i}(\hat{\boldsymbol{x}}_{ij}, t^n), \\ U_R = U_{T_{ij}}(\hat{\boldsymbol{x}}_{ij}, t^n), \end{cases} \begin{cases} U'_{\xi L} = \boldsymbol{\sigma}_i^n \cdot \boldsymbol{n}_{ij}, \\ U'_{\xi R} = \boldsymbol{\sigma}_{ij}^n \cdot \boldsymbol{n}_{ij}. \end{cases} \tag{2.6}$$

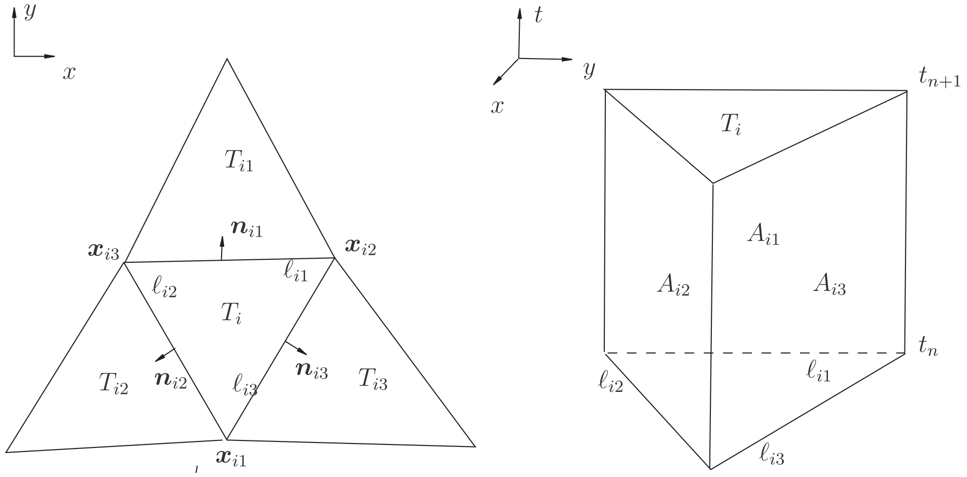


Fig. 2.1. Triangular meshes and control volume.

Note that the local GRP is solved in the exactly same way as in [20] (see Eq. (2.1) and (2.7) there). Once it is solved and the interface values $U^n_{A_{ij}}$ and $\left(\frac{\partial U}{\partial t}\right)^n_{A_{ij}}$ are obtained, we can immediately define a "centroid" value on the interface A_{ij} with the formula

$$U^{n+\frac{1}{2}}_{A_{ij}} := U^n_{A_{ij}} + \frac{\Delta t}{2}\left(\frac{\partial U}{\partial t}\right)^n_{A_{ij}}, \tag{2.7}$$

used in numerical fluxes (2.4). The value $U^n_{A_{ij}}$ is available with the standard Riemann solver [42], and the instantaneous derivative $\left(\frac{\partial U}{\partial t}\right)^n_{A_{ij}}$ is obtained by solving the generalized Riemann problem [7,20]. Then we implement the GRP solver in the following three steps:

Step 1. Given piecewise linear initial data (2.3), calculate the centroid value $U^{n+\frac{1}{2}}_{A_{ij}}(j = 1, 2, 3)$ for the every cell T_i with the formula (2.7) by solving the local generalized Riemann problem (2.5).
Step 2. Evaluate the new cell averages U^{n+1}_i using (2.2) and (2.4).
Step 3. Update the gradient $\boldsymbol{\sigma}^{n+1}_i$ with the least squares method that will be introduced in the next section.

3. Data reconstruction

The data reconstruction is a necessary step for Godunov-type or MUSCL-type schemes in order to obtain piecewise linear/polynomial reconstruction according to the desired accuracy order. If structural meshes are preferred, the gradients can be updated together with the solution evolution in the framework of the GRP methodology. However, for the present study of unstructured meshes, this GRP technique is not available. Instead, the following gradient predicting method is chosen, while the GRP merit on structured meshes is awaiting for further study over unstructured meshes.

3.1. Gradient predicting methods

There are various methods such as (weighted) least squares method [35], Green-Gauss method [1] and multislope method [10], to construct the gradient σ^n_i of solutions in order to reconstruct the initial data for (2.3). In this paper we adopt the weighted least squares method in [35].

The (weighted) least squares gradient construction is a technique independent of the mesh topology and only relies on a stencil that identifies relevant neighboring cells for use in the gradient estimation. We choose the stencil of the nearest neighboring values, i.e. the data of T_i and $T_{ij}, j = 1, 2, 3$, see Fig. 2.1. The gradient $(q_x, q_y)_i$ of a physical variable q over the triangle T_i is obtained by solving for the values of the gradient which minimize the sum of the squares of the differences between neighboring values and values extrapolated from the point $\bar{\boldsymbol{x}}_i$ under consideration to the neighboring locations $\bar{\boldsymbol{x}}_{ij}$ $(j = 1, 2, 3)$. Notations are referred to those in SubSection 2.2. More precisely, set

$$\mathcal{R}(q_x, q_y) := \sum_{j=1}^{3} w^2_{ij}[-\delta q_{ij} + (q_x, q_y)\cdot(\bar{\boldsymbol{x}}_{ij} - \bar{\boldsymbol{x}}_i)]^2, \tag{3.1}$$

then the gradient $(q_x, q_y)_i$ satisfies

$$\mathcal{R}((q_x, q_y)_i) = \min_{(q_x,q_y)\in\mathbb{R}^2}\{\mathcal{R}(q_x, q_y)\}, \tag{3.2}$$

where δq_{ij} represents the difference $q_{ij} - q_i$, and w_{ij} is a weighting factor. Here and below, q_i and q_{ij} are the quantities at the cell T_i and its neighboring cell T_{ij}, respectively. In this work, we use the inverse distance weight, i.e. $w_{ij} = 1/|\bar{\boldsymbol{x}}_{ij} - \bar{\boldsymbol{x}}_i|, \ j = 1, 2, 3$. Then the gradient $(q_x, q_y)_i$ is obtained straight-forwardly by solving the following algebraic system

$$\begin{cases} a_i(q_x)_i + b_i(q_y)_i = d_i, \\ b_i(q_x)_i + c_i(q_y)_i = e_i, \end{cases}$$

where the coefficients are

$$a_i = \sum_{j=1}^{3} w^2_{ij}\delta x^2_{ij}, \quad b_i = \sum_{j=1}^{3} w^2_{ij}\delta x_{ij}\delta y_{ij}, \quad c_i = \sum_{j=1}^{3} w^2_{ij}\delta y^2_{ij},$$
$$d_i = \sum_{j=1}^{3} w^2_{ij}\delta x_{ij}\delta q_{ij}, \quad e_i = \sum_{j=1}^{3} w^2_{ij}\delta y_{ij}\delta q_{ij},$$

and $(\delta x_{ij}, \delta y_{ij}) := \bar{\boldsymbol{x}}_{ij} - \bar{\boldsymbol{x}}_i$. An intensive discussion on least squares method can be found in [35] and references therein.

Remark 3.1. According to Haider et al. [19], for the MUSCL–type cell–centered finite volume methods on unstructured meshes, if the least squares gradient reconstruction gives an unstable scheme, then any other consistent slope reconstruction is also very likely to result in an unstable scheme. Furthermore, the gradient for the cell T_i near the boundary (i.e. there is at least one side of the cell along the boundary) is specified to zero for the sake of simplicity.

3.2. Limiters

To avoid nonphysical oscillations in the solution, some limiting strategies should be applied to correct the gradient under some physical considerations. It is difficult to extend TVD (*Total Variation Diminishing*) limiters for two– or multi–dimensional problems, especially for general (arbitrary) unstructured meshes. Nevertheless, the LED (*Local Extremum Diminishing*) principle, introduced by Jameson et al. [1,27], in which the local maximum does not increase and the minimum does not decrease, can be applied for multidimensional problems on both structured and unstructured meshes. We adopt the Barth limiter [1], except for the steady flow simulation where the Venkatakrishnan limiter [43] is applied to accelerate the convergence rate.

Assume that $q_i,\ q_{ij}$ are the same representations as those aforementioned. Denote that

$$q_i^{max} := \max_j\{q_i, q_{ij}\}, \quad q_i^{min} := \min_j\{q_i, q_{ij}\},$$

$$\hat{q}_{ij} := \hat{q}_{ij}(\hat{\boldsymbol{x}}_{ij}) = q_i + (\nabla q)_i \cdot (\hat{\boldsymbol{x}}_{ij} - \bar{\boldsymbol{x}}_i),$$

where $\nabla q = (q_x, q_y)^\top$, and denote by

$$L_{ij} = \begin{cases} \min(1, \frac{q_i^{max} - q_i}{\hat{q}_{ij} - q_i}), & \text{if } \hat{q}_{ij} - q_i > 0, \\ \min(1, \frac{q_i^{min} - q_i}{\hat{q}_{ij} - q_i}), & \text{if } \hat{q}_{ij} - q_i < 0, \\ 1, & \text{if } \hat{q}_{ij} - q_i = 0. \end{cases} \tag{3.3}$$

The Barth limiter for the element T_i is taken as

$$L_i = \min\{L_{i1}, L_{i2}, L_{i3}\}. \tag{3.4}$$

Then the gradient $(\nabla q)_i$ is updated as $(\nabla q)_i L_i$.

Remark 3.2. The Barth limiter can ensure that the reconstructed values be bounded by the values of the cell and its neighbors to avoid nonphysical oscillations effectively even on highly irregular meshes [1]. But it is not friendly to the convergence of upwind Euler and Navier–Stokes codes to steady states. For the steady simulation in Section 6.3.4, we use the more diffusive limiter of Venkatakrishnan [43], which was also given as (3.4) by replacing (3.3) with

$$L_{ij} = \begin{cases} \phi(\frac{q_i^{max} - q_i}{\hat{q}_{ij} - q_i}), & \text{if } \hat{q}_{ij} - q_i > 0, \\ \phi(\frac{q_i^{min} - q_i}{\hat{q}_{ij} - q_i}), & \text{if } \hat{q}_{ij} - q_i < 0, \\ 1, & \text{if } \hat{q}_{ij} - q_i = 0, \end{cases} \tag{3.5}$$

where the function ϕ is defined as

$$\phi(z) = \frac{z^2 + 2z}{z^2 + z + 2}, \quad z \in \mathbb{R}. \tag{3.6}$$

Remark 3.3. There are many other limiters for MUSCL-type finite volume scheme on unstructured meshes. The design and implementation of limiters, particularly multidimensional limiters, is still an active field of research [24].

4. Adaptive mesh redistribution

This section illustrates the adaptive mesh redistribution method on the variational formulation briefly. The adaptive moving mesh methods have been successfully applied in a variety of scientific and engineering areas such as fluid dynamics and solid mechanics etc., where singular or nearly singular solutions are solved dynamically in fairly localized regions. There were a lot of important progress on moving mesh methods for partial differential equations, as cited in the introduction section.

4.1. Mesh redistributions based on the variational method

The adaptive mesh redistribution can be regarded as an approximation of a coordinate transformation between computational coordinates $\zeta = (\xi, \eta) \in \Omega_c$ (with quasi-uniform mesh partition) and physical coordinates $\boldsymbol{x} = (x, y) \in \Omega_p$ (with non-uniform adaptive mesh).

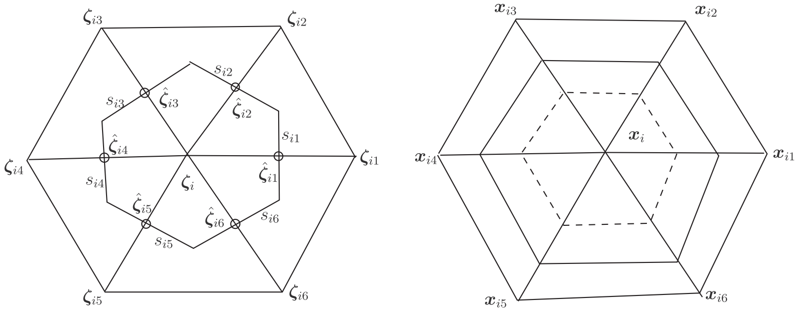


Fig. 4.1. Schematic diagram of the dual cell V_i and the triangle elements of Ω_c; (left) and Ω_p; (right).

Let Ω_c be the computational domain with the orthogonal coordinate $\zeta = (\xi, \eta)$, whose data structure is the same as that of the triangulation of the physical domain Ω_p. Let V_i denote the dual cell associated with the internal vertex ζ_i in Ω_c, which is delimited in jointing the centroid of all the triangles surrounding ζ_i. See Fig. 4.1.

We follow Chen, Tang and Zhang [13] for the adaptive process. Denote by $\aleph(i)$ the number of triangles surrounding ζ_i, in Fig. 4.1, and $\hat{\zeta}_{ij}$ the midpoint of the edge $\hat{s}_{ij}$ connecting nodes ζ_i and ζ_{ij}. Let s_{ij} be a linear segment connecting centroids of two neighboring triangles surrounding ζ_i and ζ_{ij}. Here we have assumed that s_{ij} lies on the perpendicular bisector of the edge $\hat{s}_{ij}$ and this assumption can be relaxed in practice. A one-to-one coordinate transformation from the computational domain Ω_c to the physical domain Ω_p is denoted by $\boldsymbol{x} = \boldsymbol{x}(\zeta), \zeta \in \Omega_c$. A "mesh-energy" functional defined in Ω_c is

$$\mathrm{E}(\boldsymbol{x}) = \frac{1}{2}\int_{\Omega_c} [\overline{\nabla}^\top x \mathbf{G}_1 \overline{\nabla} x + \overline{\nabla}^\top y \mathbf{G}_2 \overline{\nabla} y] d\xi d\eta, \tag{4.1}$$

where $\overline{\nabla} = (\frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta})^\top$, $\mathbf{G}_k = \omega \mathbf{I}$ $(k = 1, 2)$ in practice [45], $\mathbf{I}$ is the identity matrix, and the monitor function ω is a positive weighted function. This produces an isotropic mesh adaptation. The corresponding Euler–Lagrange equation of (4.1) is

$$\overline{\nabla} \cdot (\omega \overline{\nabla} \boldsymbol{x}) = \mathbf{0}, \tag{4.2}$$

where $\omega = \omega(\overline{\nabla} U) := \omega(U)$, to be defined in Subsection 4.4.

We give a finite volume approximation of (4.2) subject to boundary conditions

$$\boldsymbol{x} \in \partial\Omega_p, \quad \text{if } \zeta \in \partial\Omega_c.$$

The following discrete mesh equation

$$\sum_{j=1}^{\aleph(i)} \omega_{ij} |s_{ij}| \frac{\boldsymbol{x}_{ij} - \boldsymbol{x}_i}{|\hat{s}_{ij}|} = 0 \tag{4.3}$$

is used, where $\omega_{ij} = \omega(U(\hat{\zeta}_{ij})), |s_{ij}|$ and $|\hat{s}_{ij}|$ are the lengths of the edges s_{ij}, and $\hat{s}_{ij}$, respectively. Generally, (4.3) is a nonlinear algebraic system, due to dependence of ω_{ij} on the solution. To avoid this difficulty arising from the nonlinearity, we linearize (4.3) and then use a relaxed Jacobian iteration method to solve it as follows,

$$\hat{\boldsymbol{x}}_i = \sum_{j=1}^{\aleph(i)} W_{ij} \boldsymbol{x}_{ij}^{[v]} / \sum_{j=1}^{\aleph(i)} W_{ij}, \tag{4.4}$$

$$\boldsymbol{x}_i^{[v+1]} = \mu_i \hat{\boldsymbol{x}}_i + (1 - \mu_i) \boldsymbol{x}_i^{[v]}, \tag{4.5}$$

for $v = 0, 1, \ldots$, where

$$W_{ij} = \frac{\tau}{|V_i|} \omega_{ij} |s_{ij}| / |\hat{s}_{ij}|, \quad \text{and } \mu_i = \max\left\{\sum_{j=1}^{\aleph(i)} W_{ij}, \kappa\right\}.$$

Here τ and κ are two artificial parameters that control the quality of the mesh movement. Just as the same as in [13], the mesh iteration is continued until $||\boldsymbol{x}^{[v]} - \boldsymbol{x}^{[v+1]}|| \leqslant 10^{-6}$ or $v < 5, \kappa = 0.3$ and $\max_i \{\sum_{j=1}^{\aleph(i)} W_{ij}\} = 0.5$, which ensures basically that $\boldsymbol{x}_i^{[v+1]}$ will be within the convex hull of the midpoints of the edge $\boldsymbol{x}_i \boldsymbol{x}_{ij}$, $j = 1, 2, \ldots, \aleph(i)$.

4.2. Conservative interpolation on new meshes

With each iteration step of (4.4) and (4.5), we remap the approximate solutions from the old mesh $\{\boldsymbol{x}_i^{[v]}\}$ or $\{T_i^{[v]}\}$ onto the newly resulting mesh $\{\boldsymbol{x}_i^{[v+1]}\}$ or $\{T_i^{[v+1]}\}$. Let D_{ij} denote the region scanned by the edge $s_{ij}^{[v]}$ after one iteration step of (4.4) and (4.5), $j = 1, 2, 3$. See Fig. 4.2. We use the method proposed by [13] to remap the conservative variables as

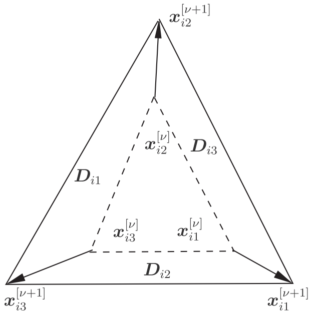


Fig. 4.2. Movement of the control volume $E_i^{[\nu]}$ to $E_i^{[\nu+1]}$.

$$|T_i^{[\nu+1]}|U_{T_i^{[\nu+1]}}^{[\nu+1]} = |T_i^{[\nu]}|U_{T_i^{[\nu]}}^{[\nu]} + \sum_{j=1}^{3} Q_{ij}, \tag{4.6}$$

where Q_{ij} is the integral of the approximate solution U over the domain D_{ij}. It can be approximately calculated as

$$Q_{ij} = \max\{|D_{ij}|, 0\}U_R + \min\{|D_{ij}|, 0\}U_L, \quad j = 1, 2, 3, \tag{4.7}$$

where U_L and U_R are reconstructed from left and right states on the edge A_{ij} (see (2.3) and (2.6)), and $|D_{ij}|$ is the signed area function of D_{ij}. Take D_{i1} in Fig. 4.2 as an example

$$|D_{i1}| = \frac{1}{2}(\boldsymbol{x}_{i2}^{[\nu]} - \boldsymbol{x}_{i2}^{[\nu+1]}) \times (\boldsymbol{x}_{i3}^{[\nu+1]} - \boldsymbol{x}_{i2}^{[\nu+1]}) + \frac{1}{2}(\boldsymbol{x}_{i3}^{[\nu+1]} - \boldsymbol{x}_{i3}^{[\nu]}) \times (\boldsymbol{x}_{i2}^{[\nu]} - \boldsymbol{x}_{i3}^{[\nu]}).$$

4.3. Reconstruction of the gradient on new meshes

After each iteration step of (4.4) and (4.5) and conservative interpolation (4.6) for conservative variables on new meshes, we should update the gradient in order to reconstruct the linear distribution. In the present work, we adopt the least squares method introduced in Section 3.1 to reconstruct gradient using the new conservative variables on the new meshes. Steps are exactly the same as those in Section 3.1.

4.4. The monitor function

The monitor function is one of the most important elements in the adaptive moving mesh algorithm. An appropriate choice of the monitor will produce grids with good quality in terms of smoothness, skewness and aspect ration [11]. In this work, we use the following monitor function,

$$\omega_{T_i} = \sqrt{1 + \alpha_\rho \tilde{\omega}_{T_i}(\beta_\rho, \rho) + \alpha_u \tilde{\omega}_{T_i}(\beta_u, u) + \alpha_v \tilde{\omega}_{T_i}(\beta_v, v) + \alpha_e \tilde{\omega}_{T_i}(\beta_e, e)}, \tag{4.8}$$

where α_q and $\beta_q \in (0, 1]$ ($q = \rho$, or u, or v, or e) are some problem-dependent parameters to be ascertained in each numerical example and $\tilde{\omega}_{T_i}$ is defined as

$$\tilde{\omega}_{T_i}(\beta_q, q) := \min\{1, |\nabla q|_{T_i}^2 / \Theta\}, \quad \Theta = \beta_q \max_{T_i}\{|\nabla q|_{T_i}^2\}. \tag{4.9}$$

5. The algorithm of adaptive GRP scheme

Now the adaptive algorithm is available with the GRP solver and it consists of two interrelated parts: The fluid dynamical evolution and the adaptive moving unstructured triangular mesh redistribution. The first part is to evolve the compressible fluid flows on fixed unstructured triangular meshes with the GRP solver, as shown in Section 2. Note that this GRP solver has second order accuracy both in space and time and is different from the classical first order exact or approximate Riemann solver [42]. In the second part, the triangular mesh points are redistributed by the relaxed Jacobian iteration method (4.4, 4.5). Then the conservative variables U are updated on the resulting new meshes by the conservative interpolation formula (4.6)–(4.7) and the least squares method is applied to reconstruct the gradient using the new conservative variables on the new meshes. This adaptive GRP scheme on unstructured triangular meshes are summarized in the following.

Step 1. At $t_n = 0$, initial quasi-uniform triangulation of the physical domain Ω_p and the computational domain Ω_c are given and they are denoted by $\{\boldsymbol{x}_i^0\}$ and $\{\zeta_i\}$ respectively. Then the cell average $U_i^{[0]}$ of the conservative variables U and the gradient $\boldsymbol{\sigma}_i^{[0]}$ are computed.
As $t_n > 0$, set $\boldsymbol{x}_i^{[0]} := \boldsymbol{x}_i^n, U_i^{[0]} := U_i^n$, and $\boldsymbol{\sigma}_i^{[0]} := \boldsymbol{\sigma}_i^n$.

Step 2. For $\nu = 0, 1, \ldots, \mu - 1$, we do the following steps:
(a) Remove the mesh points $\boldsymbol{x}^{[\nu]}$ to $\boldsymbol{x}^{[\nu+1]}$ by solving (4.4, 4.5);
(b) Update the conservative variables $U_i^{[\nu+1]}$ on the new mesh $\{\boldsymbol{x}_i^{[\nu+1]}\}$ according to (4.6) and (4.7);
(c) Adopt the least squares method to reconstruct the gradient $\boldsymbol{\sigma}_i^{[\nu+1]}$, using the new conservative variables $U_i^{[\nu+1]}$ on the new meshes $\{\boldsymbol{x}_i^{[\nu+1]}\}$;
(d) Repeat steps (a), (b) and (c) for a fixed number μ until $\sum_i |\boldsymbol{x}_i^{[\nu+1]} - \boldsymbol{x}_i^{[\nu]}| \leqslant \varepsilon$.
Step 3. Set $\boldsymbol{x}_i^n := \boldsymbol{x}_i^{[\nu+1]}, U_i^n := U_i^{[\nu+1]}, \boldsymbol{\sigma}_i^n := \boldsymbol{\sigma}_i^{[\nu+1]}$, and evolve the governing Eqs. (2.1) on the adaptive meshes $\{\boldsymbol{x}_i^n\}$ by using the GRP scheme in Section 2 to obtain the numerical approximation U_i^{n+1} at time $t = t_{n+1}$.
Step 4. Go to **Step 1** if $t_{n+1} < T$.

6. Numerical examples

In order to validate the performance of the current algorithm, four types of examples are examined. We first use two special cases to test the accuracy of the (adaptive) GRP scheme on unstructured triangular meshes. Then we use the two-dimensional algorithm proposed here to check two one-dimensional examples. The genuinely two-dimensional examples are two-dimensional Riemann problems, the double Mach reflection and forward facing step problems. Finally we implement the algorithm over the classical airfoil problem. The AGRP results are compared with those by the classical Godunov scheme and the GRP scheme. In all examples the initial mesh is generated by the free software EASYMESH, and the polytropic index γ is taken as 1.4. Take notions the same as Section 2.2, and let $\mathcal{N}(i)$ be the set whose elements are the index corresponding to the cell T_{ij} neighboring to the cell T_i. We use the following CFL-condition for the scheme (2.2) on triangular meshes shown partly in Fig. 2.1:

$$\sup_{i\in I} \frac{\Delta t_n}{|T_i|} \max_{j\in\mathcal{N}(i)} \lambda_{ij} \leqslant \mathrm{CFL}$$

with $\lambda_{ij} = (|\boldsymbol{u}\cdot\boldsymbol{n}_{ij}| + c)|\ell_{ij}|$, where $\boldsymbol{u} = (u, v)^\top, c = \sqrt{\frac{\partial p(\rho,S)}{\partial \rho}}|_S$ are the approximated velocity vector and sound speed on the triangle T_i at the time $t = t_n$, the Courant number CFL = 0.45 unless stated exceptionally.

6.1. Accuracy test

We choose two special cases to test the accuracy of the present scheme over unstructured meshes. The first only involves one rarefaction wave in the solution with initial data,

$$(\rho, u, v, p)(x, y, 0) = \begin{cases} (0.5686, & 0.3, & 0.0, & 0.3302), & x > 0.5, \\ (1.0, & -0.244, & 0.0, & 7279), & x < 0.5. \end{cases}$$

The L^2 error and the corresponding accuracy order of the (adaptive) GRP scheme are shown in Table 1. The accuracy of the AGRP scheme is obviously higher than that of the GRP scheme and can be of second order.

The second test case only involves one shock with initial data,

$$(\rho, u, v, p)(x, y, 0) = \begin{cases} (0.125, & 0.0, & 0.0, & 0.1), & x > 0.5, \\ (0.2328, & 0.7349, & 0.0, & 0.2458), & x < 0.5. \end{cases}$$

In Table 2, we see that the accuracy order is only of the first order, but the L^2 error the AGRP scheme is much smaller than that of the GRP scheme. It can be also observed in the numerical examples as follows that the resolution of the AGRP scheme is much better than that of the GRP scheme.

6.2. One-dimensional examples

We compute two well-understood one–dimensional examples but in a two–dimensional domain $\Omega_p := [0, 1] \times [0, 0.1]$ to validate the performance of the present adaptive GRP scheme. An initial quasi-uniform triangulation of physical domain Ω_p as well as the computational domain Ω_c is generated by 1288 nodes, 3641 sides and 2354 cells, whose side length is approximately equal to 1/100.

On the bottom and top boundaries are specified reflective boundary conditions; and on the left and right boundaries are imposed transmissive boundary conditions (see [42] Page 224). These one-dimensional results are displayed by extracting

Table 1
The L^2 error and the accuracy order for the single rarefaction wave case.

Methods	$h_0(=1/25)$	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
GRP	$6.59e-4(-)$	$2.41e-4(1.45)$	$7.08e-5(1.77)$	$1.89e-5(1.91)$	$4.83e-6(1.97)$
AGRP	$5.14e-4(-)$	$1.55e-4(1.73)$	$4.35e-5(1.83)$	$1.10e-5(1.98)$	$2.59e-6(2.09)$

Table 2
The L^2 error and the accuracy order for the single shock wave case.

Methods	$h_0(=1/25)$	$h_0/2$	$h_0/4$	$h_0/8$	$h_0/16$
GRP	$3.70e-2(-)$	$2.13e-2(0.80)$	$1.12e-2(0.93)$	$5.81e-3(0.95)$	$3.03e-3(0.93)$
AGRP	$2.74e-2(-)$	$1.58e-2(0.79)$	$8.62e-3(0.87)$	$4.51e-3(0.93)$	$2.20e-3(1.03)$

the data along the central horizontal line $y = 0.05$ at time $T = 0.15$. These two examples are the Sod problem and the low density problem as follows.

6.2.1. The Sod problem

The first example is the Sod shock tube problem [40] with the Riemann initial data,

$$(\rho, u, v, p)(x, y, 0) = \begin{cases} (0.125, & 0.0, & 0.0, & 0.1), & x > 0.5, \\ (1.0, & 0.0, & 0.0, & 1.0), & x < 0.5. \end{cases}$$

The numerical result is illustrated in Fig. 6.1 and the parameters in the monitor function are taken as $\alpha_\rho = 1.8, \beta_\rho = 0.01, \alpha_q = \beta_q = 0$ $(q = e, u, v)$. In comparison with the corresponding exact solutions, the difference of three schemes is quit clear. The results are improved obviously by the GRP scheme. The adaptive GRP scheme suppresses oscillations effectively and does the best in capturing these three elementary waves, see Fig. 6.1. The adaptive meshes at $T = 0.15$ for the AGRP scheme are illustrated in the Fig. 6.2.

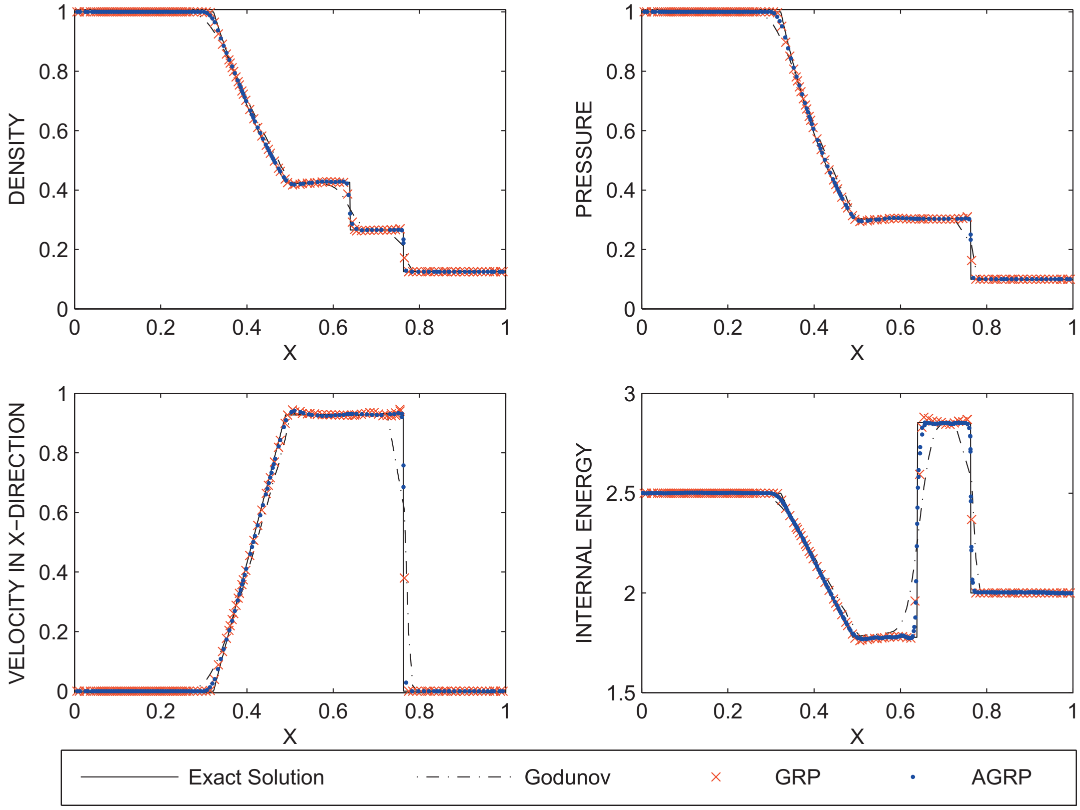


Fig. 6.1. Numerical results for the Sod problem.

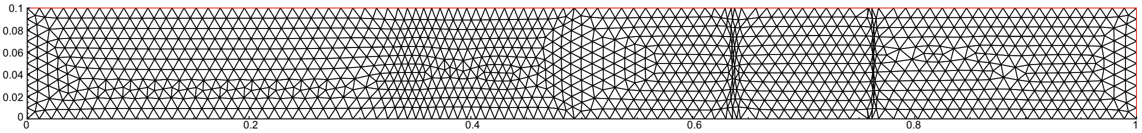


Fig. 6.2. The adaptive meshes at $T = 0.15$ for the Sod problem.

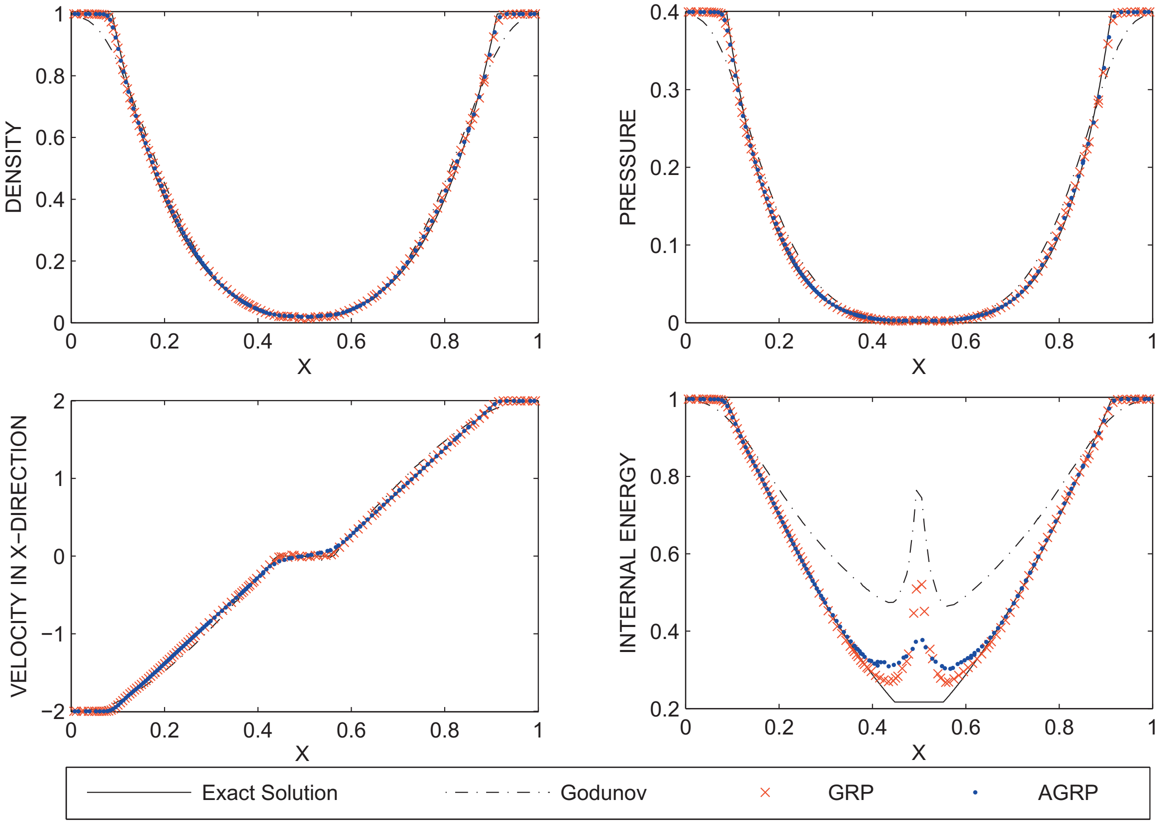


Fig. 6.3. Numerical results for the low density problem.

6.2.2. The low density problem

Initially, the data are chosen as

$$(\rho, u, v, p)(x, y, 0) = \begin{cases} (1.0, & 2.0, & 0.0, & 0.4), & x > 0.5, \\ (1.0, & -2.0, & 0.0, & 0.4), & x < 0.5. \end{cases}$$

The result is shown in Fig. 6.3 and the parameters in the monitor function are taken as $\alpha_\rho = \alpha_e = 1.8, \beta_\rho = \beta_e = 0.01, \alpha_q = \beta_q = 0\,(q = u, v)$. Note that in the internal energy plots there are a bit more errors in the interval $[0.4, 0.6]$ in which the density is too small, but the AGRP scheme preserves the positivity of the density and the internal energy well and gives better results than those in many other literatures, e.g. [10]. The adaptive meshes at $T = 0.15$ for the AGRP scheme are illustrated in the Fig. 6.4.

6.3. Two-dimensional examples

In the following we first use two well-known benchmarks, forward step problem and double Mach reflection, to test the performance of the AGRP scheme. After that, we perform the proposed AGRP scheme for two typical two-dimensional Riemann problems, which involve more complex phenomena and can be also used to test the performance of a scheme. Finally, we use it for steady problems. Although the AGRP scheme can improve the resolution of the shock but it seems that it does not give a good convergence rate, an *ad hoc* technique was used to accelerate the convergence rate.

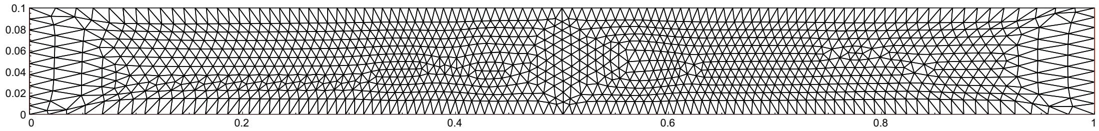


Fig. 6.4. The adaptive meshes at $T = 0.15$ for the low density problem.

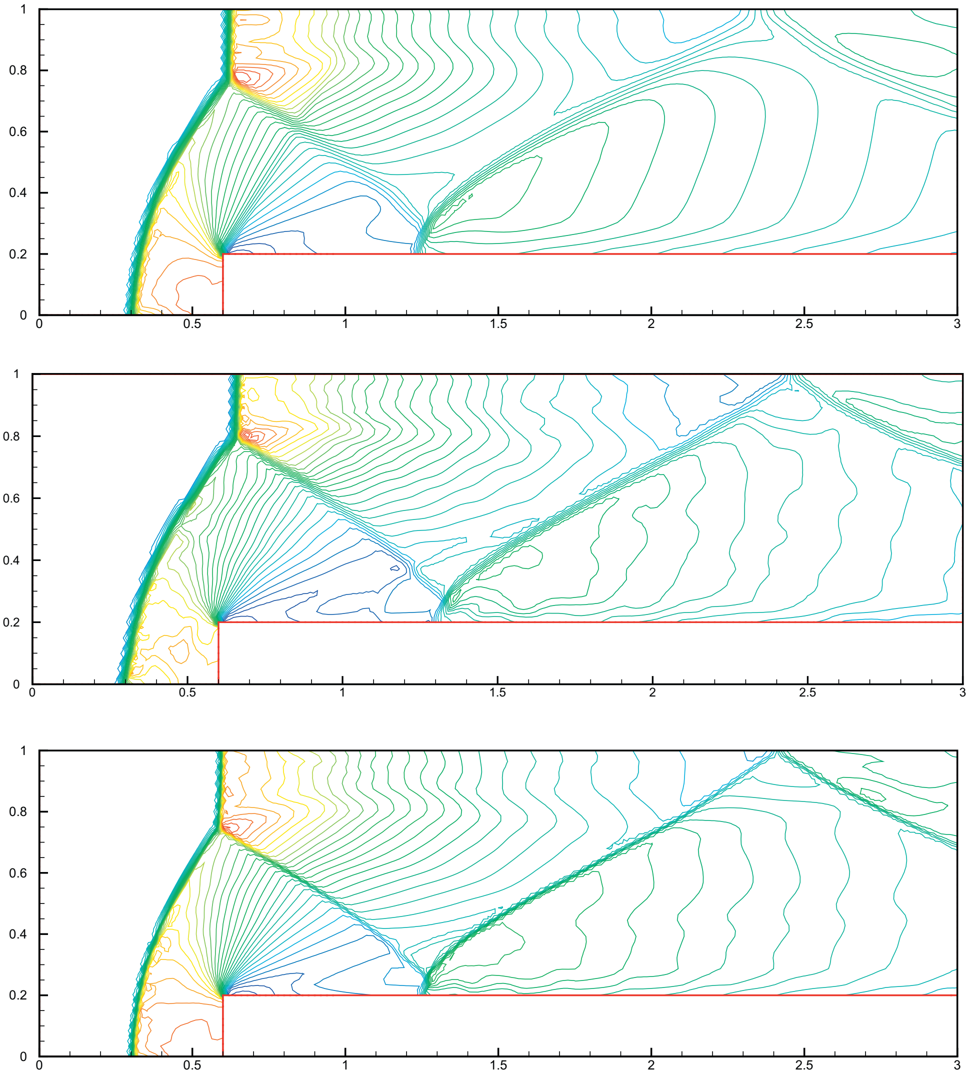


Fig. 6.5. 30 density contours of forward facing step problem using three schemes: the Godunov scheme (top), the GRP scheme (middle) and the AGRP scheme (bottom).

6.3.1. Forward facing step problem

This problem was first introduced by Emery [17], and then considered extensively by many others, e.g. [13,46]. We use exactly the same setup as in [46], i.e. the same initial and boundary conditions. The problem begins with uniform Mach 3.0 flow in a wind tunnel containing a step. The wind tunnel is 1 unit wide and 3 units long. The step is 0.2 units high and is located 0.6 units from the left-hand end of the tunnel. Initially the wind tunnel is filled with a gamma-law gas, with $\gamma = 1.4$, which has density 1.4, pressure 1.0, and velocity 3.0 everywhere. Along the walls of the tunnel reflective boundary conditions are applied. The inflow and outflow boundary conditions are specified at the left– and right–hand ends of the tunnel.

An initial quasi-uniform triangulation of Ω_p is given in the same way as that of the computational domain Ω_c and is generated 7473 nodes, 22016 sides and 14544 cells, whose edge length is approximately equal to 1/50. The parameters in the monitor function are taken as $\alpha_\rho = 1.0, \beta_\rho = 0.01, \alpha_q = \beta_q = 0$ $(q = e, u, v)$, and the adaptive mesh at $T = 4.0$ is displayed in Fig. 6.6.

The results obtained by using the AGRP scheme are improved obviously than that by the Godunov scheme on the same initial meshes. The corner of the step is the center of a rarefaction fan and hence is a singular point of the flow. Unlike [46], we do not modify the schemes near the corner at all. It is well known that this may lead to an erroneous entropy layer at the downstream bottom of the wall, as well as a spurious Mach stem at the bottom wall. However, these numerical artifacts decrease as the mesh is refined. In our work, mesh points near the corner of the step are redistributed through the adaptive procedure, which makes the result better in comparison with that by the GRP scheme on quasi-uniform meshes without this technique. See Fig. 6.5.

6.3.2. Double Mach reflection

This problem was studied extensively by Woodward and Collela [46] and later by many others, e.g. [13,20,46]. We use exactly the same setup as in [46], i.e. the same initial and boundary conditions as well as the same physical 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^o angle with the x-axis. More precisely, the initial data are

$$U(x,y,0) = \begin{cases} U_L, & y \geqslant h(x,0), \\ U_R, & \text{otherwise}, \end{cases} \tag{6.1}$$

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

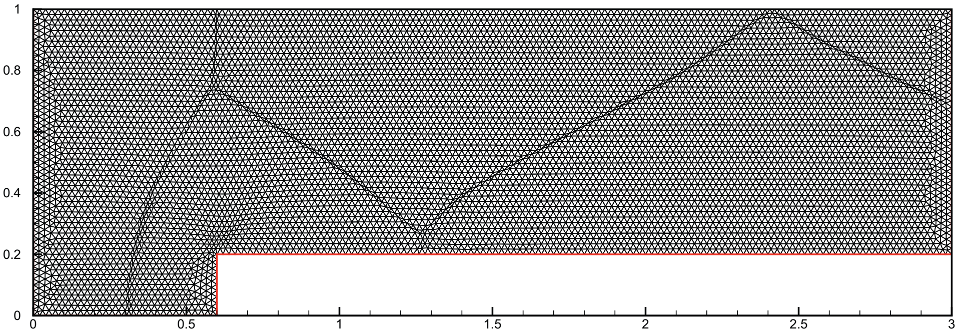


Fig. 6.6. The adaptive meshes at $T = 4.0$ for forward facing step problem.

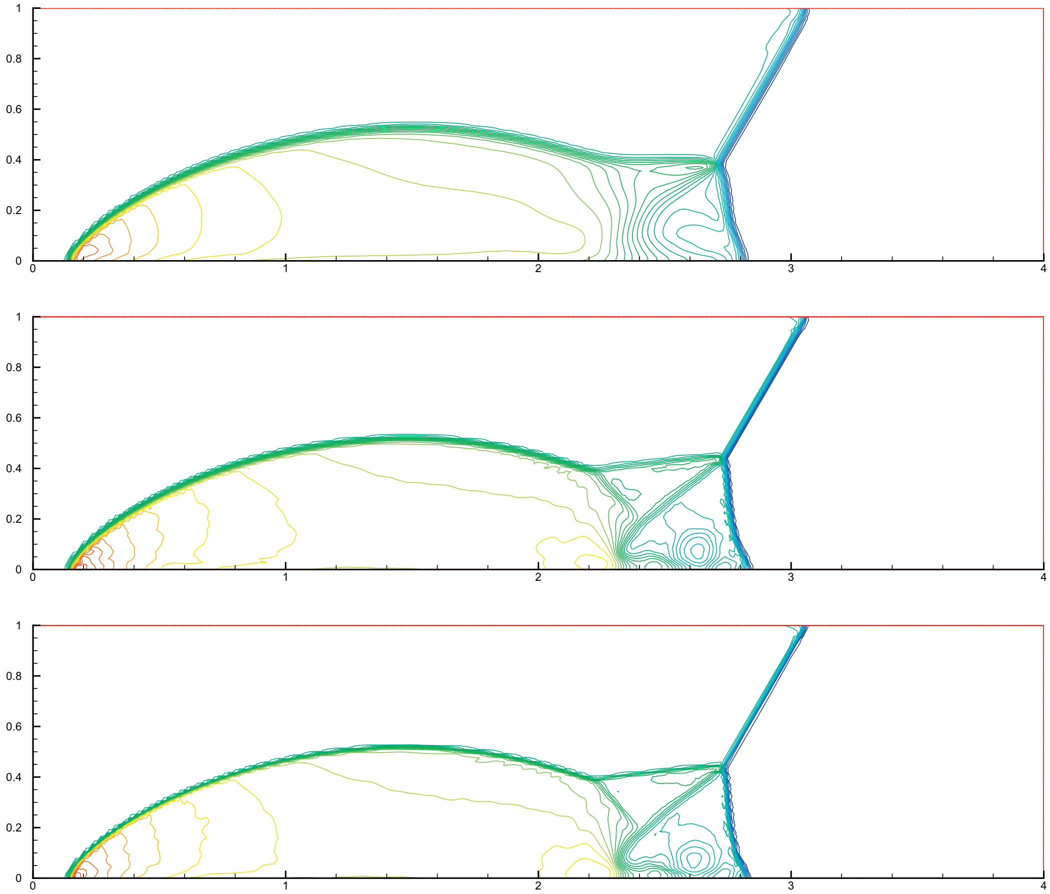


Fig. 6.7. 30 density contours of double Mach reflection problem using three schemes on the triangular mesh with 33701 cells: the Godunov scheme (top), the GRP scheme (middle) and the AGRP scheme (bottom).

$$\begin{cases} U_L = (8.0, 57.1579, -33.0012, 563.544)^\top, \\ U_R = (1.4, 0.0, 0.0, 2.5)^\top, \\ h(x,t) = \sqrt{3}(x - \frac{1}{6}) - 20t. \end{cases} \tag{6.2}$$

The reflective boundary condition is specified on the wall (the part from $x = 1/6$ to $x = 4$ of the bottom boundary), while for the rest of the bottom boundary (the part from $x = 0$ to $x = 1/6$), the exact post-shock condition is imposed. At the top boundary, the flow values are set to describe the exact motion of the Mach 10 shock. The inflow and outflow boundary conditions are specified on the left and right boundaries, respectively.

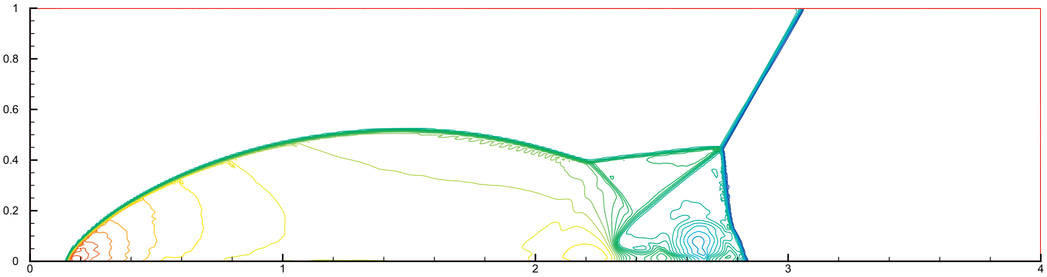


Fig. 6.8. 30 density contours of double Mach reflection problem using the GRP scheme on the triangular mesh with 134447 cells.

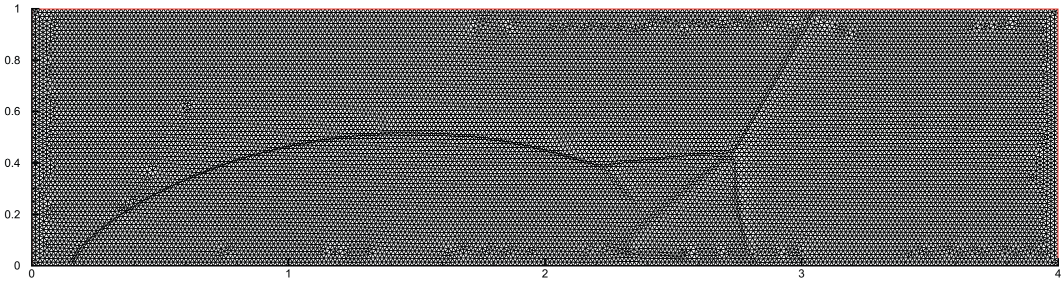


Fig. 6.9. The adaptive meshes at $T = 0.2$ for double Mach reflection problem.

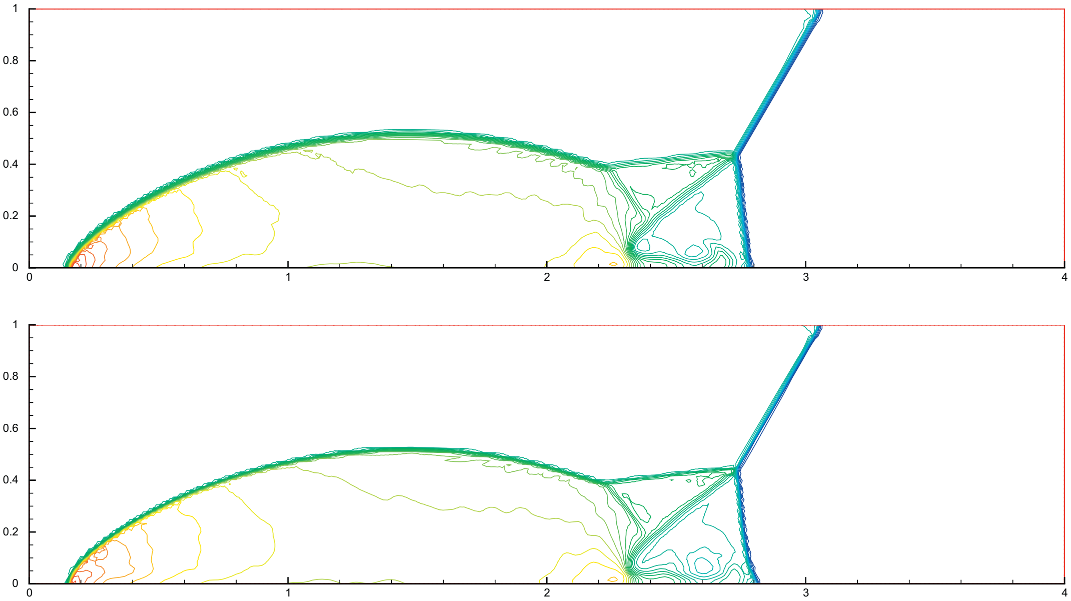


Fig. 6.10. 30 density contours of double Mach reflection problem using: the GRP scheme rescued by HLL solver (top), and the AGRP scheme rescued by HLL solver (bottom).

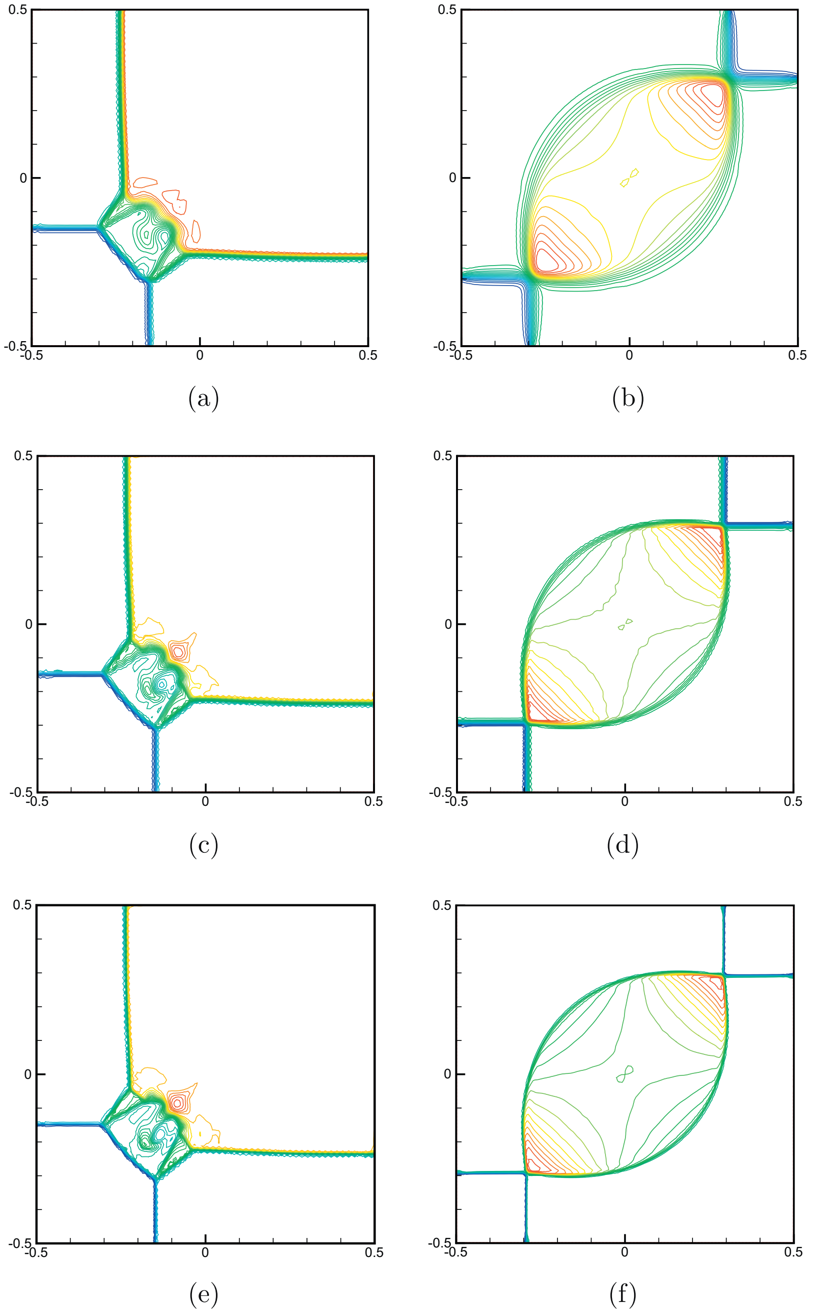


Fig. 6.11. 30 density contours for four shocks problems are displayed. Top: the Godunov scheme; middle: the GRP scheme; bottom: the AGRP scheme.

An initial quasi-uniform triangulation of Ω_p is given in the same way as that of the computational domain Ω_c and is generated 17152 nodes, 50852 sides and 33701 cells, whose edge length is approximately equal to $1/60$. To appropriately discretize the initial data, we add an internal boundary along the initial discontinuities to generate the initial mesh. The parameters in the monitor function are taken as $\alpha_\rho = 1.0, \beta_\rho = 0.02, \alpha_q = \beta_q = 0\,(q = e, u, v)$, and the adaptive mesh is shown in the Fig. 6.9.

Numerical results at time $T = 0.2$ are displayed in Figs. 6.7 and 6.8. Both the GRP scheme and the adaptive GRP scheme captured the large scale as well as small scale of the double Mach reflections and reproduced those in [20,46]. The results of the AGRP scheme are improved obviously than that of the Godunov scheme on the same initial meshes (with 33701 cells), and comparable to that of the GRP scheme on uniformly refined meshes with 134447 cells. Furthermore, the cost of the former (1221s) is less than half of the latter (2595s).

Remark 6.1. Note that the right Mach stem bend a little, which is commonly observed in the numerical results with unstructured mesh methods even with very fine meshes, e.g. Figs. 6.3, 6.4 in [38] and Figs. 4–6 in [50]. It is probably produced by the interplay between the dynamics of the solver and the topology of the mesh. However, this defect can be rescued with HLL solver, or structured meshes methods [20]. Actually, Fig. 6.10 shows the results by using the combination of the (adaptive) GRP solver and HLL solver. More precisely, in some steps (e.g. in the even time steps), the local GRP (2.5) was solved approximately by using the HLL solver. In this way, the Mach stem is better resolved in some sense at the cost of a little bit degeneration of accuracy.s

6.3.3. Simulations of two-dimensional Riemann problems

We chose two examples of 2–D Riemann problems to test our scheme. They can be found in [21,33,39,51]. Systematic treatments can be found in [30]. The notations and initial data are given the same as [21]. An initial quasi-uniform triangulation of Ω_p is given in the same way as that of the computational domain Ω_c and is generated 11830 nodes, 35087 sides and 23258 cells, whose side length is approximately equal to 1/100. The initial mesh was specially handled along the initial discontinuities to discretize the initial data more appropriately. In order to offset the numerical error due to the degenerate accuracy of the boundary cells, the exact solution is specified to the fictitious cells along the boundary. The 30 density contours are shown in the examples.

Fig. 6.11 illustrates the interaction of shocks resolved by the Godunov scheme, the GRP scheme and the adaptive GRP scheme, respectively. The two–dimensional Riemann initial data for the left column are

$$(\rho,u,v,p)(x,y,0)=\begin{cases}(1.5,0.0,0.0,1.5), & x>0,\quad y>0;\\(0.5323,1.206,0.0,0.3), & x<0,\quad y>0;\\(0.138,1.206,1.206,0.029), & x<0,\quad y<0;\\(0.5323,0.0,0.0,0.3), & x>0,\quad y<0.\end{cases}$$

The initial data for the right column are

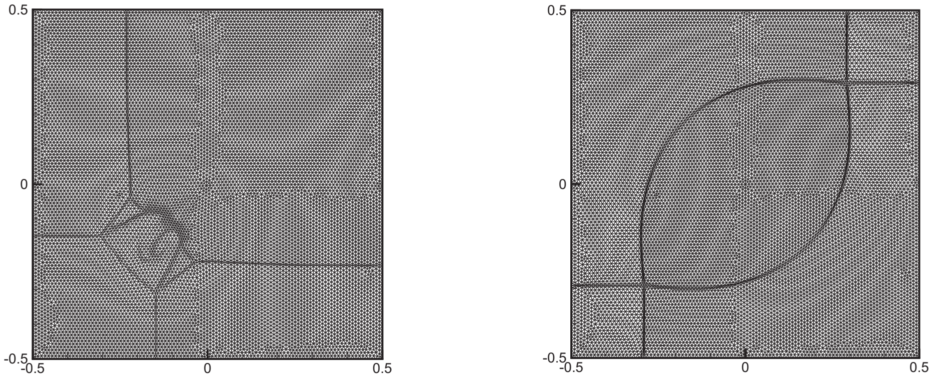
$$(\rho,u,v,p)(x,y,0)=\begin{cases}(1,-0.189970,-0.189970,1.4), & x>0,\quad y>0;\\(1.3,0.189970,-0.189970,2.025532), & x<0,\quad y>0;\\(1,0.189970,0.189970,1.4), & x<0,\quad y<0;\\(1.3,-0.189970,0.189970,2.025532), & x>0,\quad y<0.\end{cases}$$

The left column is a Mach–like interaction of shocks at $T=0.35$, and the parameters in the monitor function are $\alpha_\rho=1.0,\beta_\rho=0.01,\alpha_q=\beta_q=0.0,q=e,u,v$. The right column is a regular–like interaction of shocks at $T=0.2$, and the parameters in the monitor function are $\alpha_\rho=1.5,\beta_\rho=0.01$, $\alpha_q=\beta_q=0,q=e,u,v$. The adaptive meshes at the final computational time are shown in the Fig. 6.12. It is found that the proposed AGRP scheme capture the shocks quite sharply.

Remark 6.2. The symmetry of the solution can not be preserved perfectly if the (asymmetric) unstructured mesh method is used, which is a common disadvantage of unstructured mesh methods. See the left column of Fig. 6.11 or Figs. 23–27 in [23].

6.3.4. Steady flow simulations around the NACA0012 airfoil

In practice, unsteady algorithms are often used to simulate steady flows. The same idea is applied here for the simulation of flows around airfoil. We choose the often–used NACA 0012 airfoil. In the computations the more diffusive limiter of



(a) The adaptive meshes at $T=0.35$ for the Mach–like interaction of shocks.

(b) The adaptive meshes $T=0.2$ for the regular–like interaction of shocks.

Fig. 6.12. The adaptive meshes at the final computational time for four shocks problems are displayed.

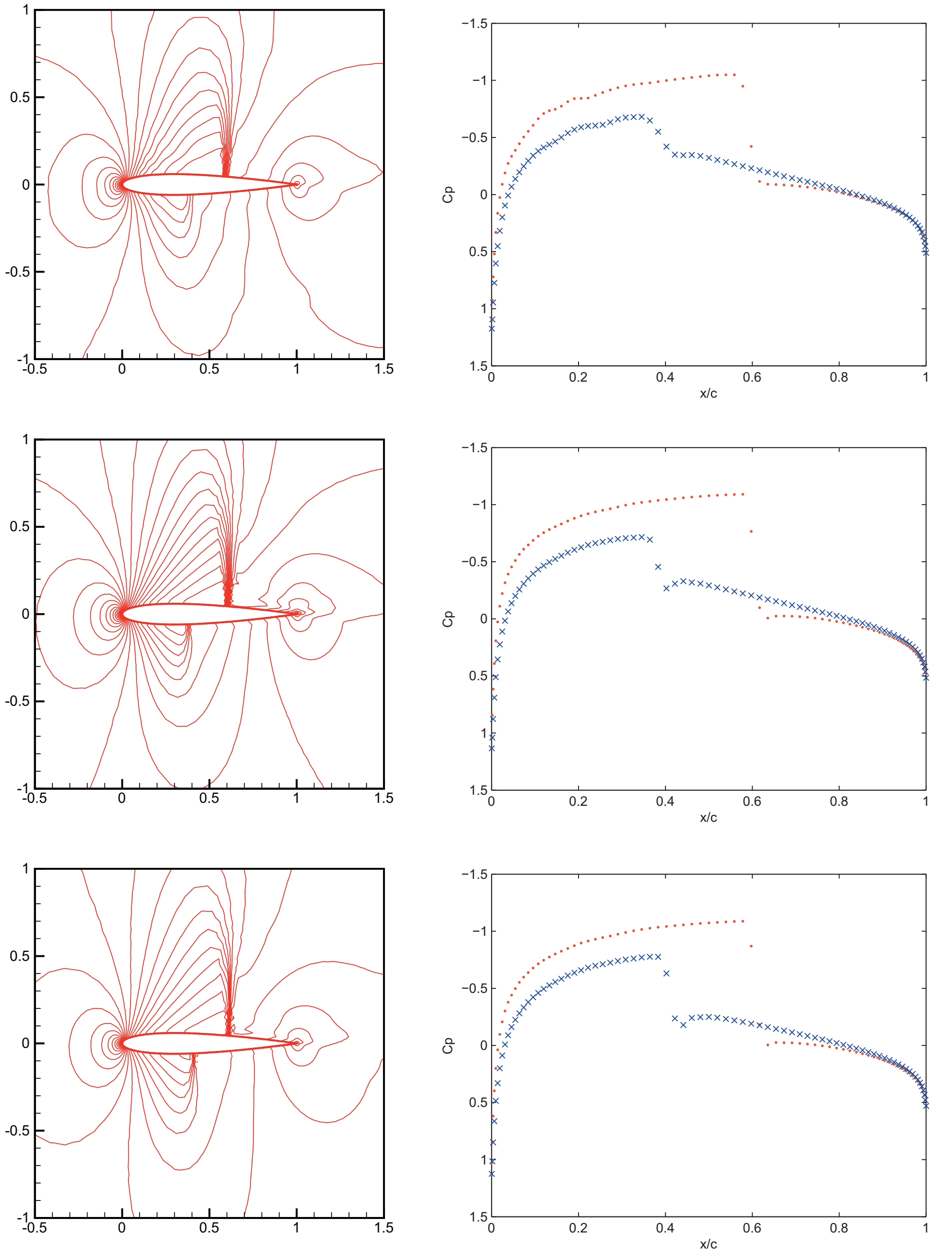


Fig. 6.13. Numerical results of transonic flow around NACA 0012 airfoil at $M_\infty = 0.8, \alpha = 1.25^\circ$ using three schemes: the Godunov scheme (top), the GRP scheme (middle) and the adaptive GRP scheme (bottom). The left column are 30 contours of Mach number, and the right column are pressure distributions in terms of pressure coefficient along the airfoil.

Venkatankrishnan [43] is applied to improve the convergence rate, and the constant CFL is taken as 0.95. The initial mesh is generated by 6016 nodes, 17762 sides and 11746 cells.

The first example has the Mach number $M_\infty = 0.8$ and the angle of attack $\alpha = 1.25^\circ$. The result is displayed in Fig. 6.13. The poor accuracy by the Godunov scheme is clearly observed. In this case, the lower surface shock is absent. The second order GRP scheme with piecewise linear reconstructions easily captures both upper and lower surface shocks. And the proposed adaptive GRP scheme resolve the shocks more sharply. The pressure coefficient plots show the excellent agrement with many literatures, e.g. [1].

The second example has the Mach number $M_\infty = 0.85$ and the angle of attack $\alpha = 0^\circ$. As shown in Fig. 6.14, the resolution difference of three schemes is quite clear, and the symmetry is well preserved by using the above three methods. It is observed that our algorithms do quite well.

Note that boundary conditions are enforced as follows. The solid boundary conditions are specified on the cells along the surface of the airfoil, and the far field boundary conditions [26] are applied to the outer boundary. Since the mesh size

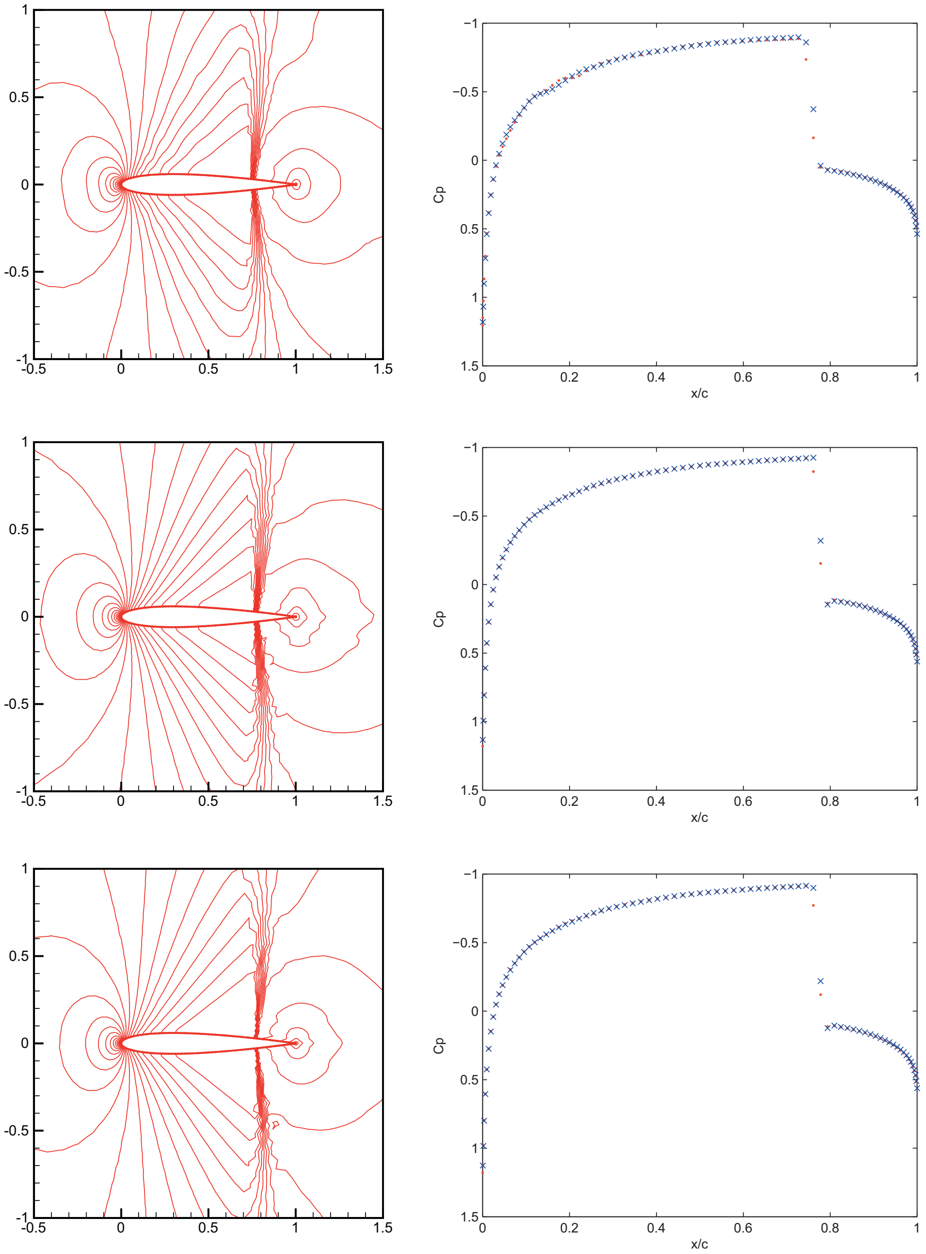
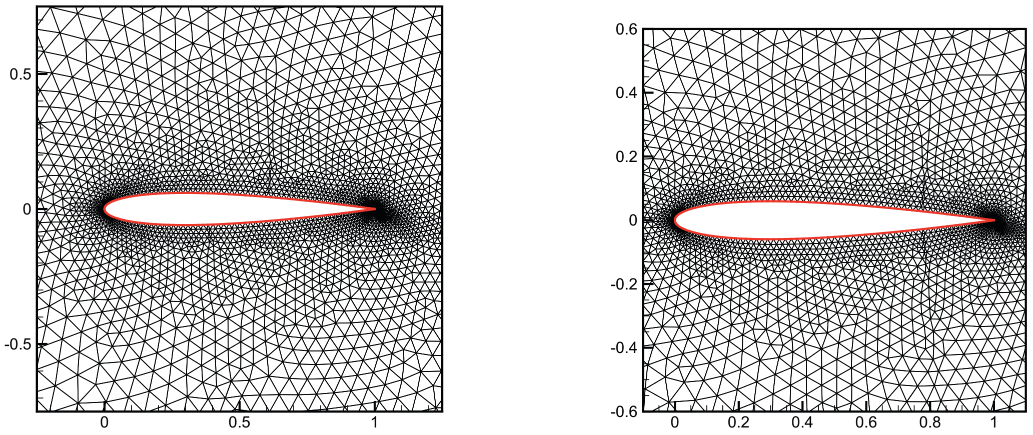


Fig. 6.14. Numerical results of transonic flow around NACA 0012 airfoil at $M_\infty = 0.85, \alpha = 0.0^\circ$ using three schemes: the Godunov scheme (top), the GRP scheme (middle) and the adaptive GRP scheme (bottom). The left column are 30 contours of Mach number, and the right column are pressure distributions in terms of pressure coefficient along the airfoil.

changes drastically through the whole physical domain due to its special geometry and high resolution requirement of the singularity of the underlying problems by using the adaptive techniques, the global time step should be kept very small, this will impair the convergence rate of the numerical method. At this point, we use the local time stepping method to accelerate the convergence rate. Specifically, we use the standard AGRP procedure to evolve the solution in the first 5,000 steps and then use the local CFL condition

$$\frac{\Delta t_j^n}{c_j^n + |u_j^n| + |v_j^n|} \leqslant 0.5 \tag{6.3}$$

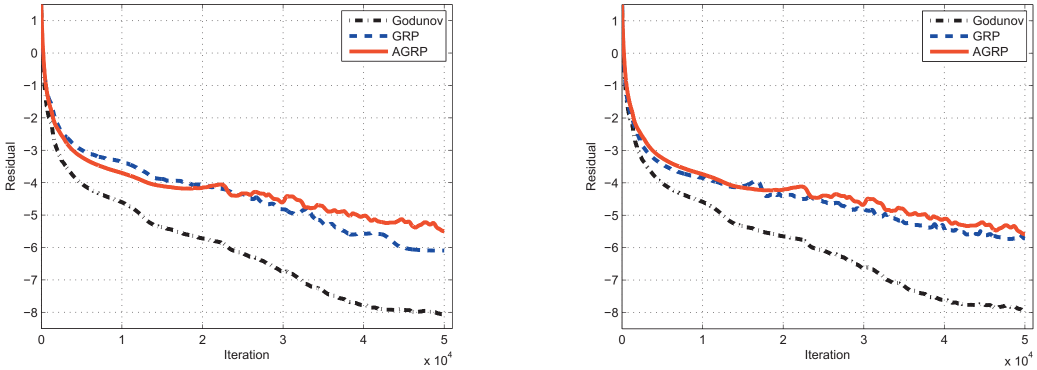
over the cell C_j to control the local time step size t_j^n at $t = t_n$. Since the flow is almost steady after 5,000 steps, this local time stepping technique is valid. Figs. 6.15,6.16 illustrate the adaptive meshes at the steady state and the convergence histories



(a) $M_\infty = 0.8, \alpha = 1.25°$. Parameters in the monitor function are taken as $\alpha_\rho = 0.8, \beta_\rho = 0.05, \alpha_e = 0.5, \beta_e = 0.1,\ \alpha_q = \beta_q = 0\ (q = u, v)$.

(b) $M_\infty = 0.85, \alpha = 0.0°$. Parameters in the monitor function are taken as $\alpha_\rho = 0.8, \beta_\rho = 0.05, \alpha_e = 0.5, \beta_e = 0.1,\ \alpha_q = \beta_q = 0\ (q = u, v)$.

Fig. 6.15. Closeup of the adaptive meshes at the steady state for the transonic flow around NACA 0012 airfoil.



(a) Convergence histories of the residual for the density. $M_\infty = 0.8, \alpha = 1.25°$.

(b) Convergence histories of the residual for the density. $M_\infty = 0.85, \alpha = 0.0°$.

Fig. 6.16. Convergence histories of the residual of the density for the transonic flow around NACA 0012 airfoil.

for the residual of density, defined as the root mean square value of $\frac{\partial\rho}{\partial t}$ [25]. In order to further ameliorate the convergence rate, some other techniques such as using multigrid methods and adding enthalpy damping [25] could also be applied.

7. Discussions

This paper combines the direct Eulerian GRP solver (of second order accuracy) with the adaptive moving mesh technique on unstructured triangular meshes to compute compressible fluid flows. The advantages of unstructured meshes are obvious in comparison with structured meshes: It is more feasible to deal with computational domains of complex geometries and easier to be incorporated into adaptive algorithms. The GRP solver also exhibits its robustness and good resolution in resolving discontinuities (shocks or vortices). Although we just work on triangular meshes in the present paper, such a algorithm can be extended to arbitrary mixtures of various unstructured polyhedral meshes.

Together with [20], we have applied the direct Eulerian GRP solver on structured and unstructured meshes. We conclude:

(1) Flexibility of the GRP solver. The second order GRP solver, just like the first order exact or approximate Riemann solver [42], can be incorporated with any techniques in the framework of finite volume to compute compressible fluid flows.

(2) Resolution and efficiency. the proposed AGRP scheme on unstructured/structured meshes can capture discontinuities very sharply. The adaptivity of meshes makes the scheme more efficient.
(3) Shock layers or/and boundary layer regions. Probably due to the presence of (numerical) boundary layers, the interplay between the dynamics of solvers and mesh topology is very crucial. In the problem of double Mach reflection, the shock is bent near the boundary when the Godunov–type solvers are applied. However, this defect can be rescued with the HLL solver or structured meshes [20].

Acknowledgement

Both authors appreciate Professor Huazhong Tang for his illuminating discussions and his generosity to teach us the adaptive moving mesh methods. The first author is partially supported by NSFC (91130021, 11031001), and also supported by an open project from Institute of Applied Physics and Computational Mathematics, Beijing.

References

[1] T. Barth, D.C. Jesperson, The design and application of upwind schemes on unstructured meshes, AIAA Report, 89-0366, 1989.
[2] M. Ben-Artzi, J. Falcovitz, A second-order Godunov-type scheme for compressible fluid dynamics, J. Comput. Phys. 55 (1) (1984) 1–32.
[3] M. Ben-Artzi, J. Falcovitz, An upwind second-order scheme for compressible duct flows, SIAM J. Sci. Stat. Comput. 7 (3) (1986) 744–768.
[4] M. Ben-Artzi, The generalized Riemann problem for reactive flows, J. Comput. Phys. 81 (1989) 70–101.
[5] M. Ben-Artzi, J. Falcovitz, Generalized Riemann Problems in Computational Gas Dynamics, Cambridge University Press, 2003.
[6] M. Ben-Artzi, J. Li, Hyperbolic balance laws: Riemann invariants and the generalized Riemann problem, Numer. Math. 106 (2007) 369–425.
[7] M. Ben-Artzi, J. Li, G. Warnecke, A direct Eulerian GRP scheme for compressible fluid flows, J. Comput. Phys. 218 (2006) 19–34.
[8] J.U. Brackbill, J.S. Saltzman, Adaptive zoning for singular problems in two dimensions, J. Comput. Phys. 46 (1982) 342–368.
[9] J.U. Brackbill, An adaptive grid with directional control, J. Comput. Phys. 108 (1993) 38–50.
[10] T. Buffard, S. Clain, Monoslope and multislope MUSCL methods for unstructured meshes, J. Comput. Phys. 229 (2010) 3745–3776.
[11] 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.
[12] 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.
[13] G.X. Chen, H.Z. Tang, P.W. Zhang, Second-order accurate Godunov scheme for multicomponent flows on moving triangular meshes, J. Sci. Comput. 34 (2008) 64–86.
[14] A. van Dam, P.A. Zegeling, A robust moving mesh finite volume method applied to 1-D hyperbolic conservation laws from magnetohydrodynamics, J. Comput. Phys. 216 (2006) 526–546.
[15] 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.
[16] Y. Di, R. Li, T. Tang, P.W. Zhang, Moving mesh finite element methods for the incompressible Navier–Stokes equations, SIAM J. Sci. Comput. 26 (2005) 1036–1056.
[17] A.F. Emery, An evaluation of several differencing methods for inviscid fluid flow problem, J. Comput. Phys. 2 (1968) 306–331.
[18] S.K. Godunov, A finite difference method for the numerical computation and discontinuous solutions of the equations of fluid dynamics, Mat. Sb. 47 (1959) 271–295.
[19] F. Haider, J.P. Croisile, B. Courbet, Stability analysis of the cell centered finite-volume MUSCL method on unstructured grids, Numer. Math. 113 (2009) 555–600.
[20] E. Han, J. Li, H. Tang, An adaptive GRP scheme for compressible fluid flows, J. Comput. Phys. 229 (2010) 1448–1466.
[21] E. Han, J. Li, H. 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.
[22] J.Q. Han, H.Z. Tang, An adaptive moving mesh method for multidimensional ideal magnetohydrodynamics, J. Comput. Phys. 220 (2007) 791–812.
[23] R.E. Harris, Z.J. Wang, High–order adaptive quadrature–free spectral volume method on unstructured, Comput. Fluids 38 (2009) 2006–2025.
[24] M.E. Hubbard, Multidimensional slope limiters for MUSCL–type finite volume schemes on unstructured grids, J. Comput. Phys. 155 (1999) 54–74.
[25] A. Jameson, W. Schmidt, E. Turkel, Numerical solution of the Euler equations by finite volume methods using Runge–Kutta time–stepping schemes, AIAA Paper 81–1259, in: AIAA 14th Fluid and Plasma Dynamic Conference, Palo Alto, June 1981.
[26] A. Jameson, Solution of the Euler equations for two dimensional transonic flow by a multigrid method, MAE Report, 1613 (1983).
[27] A. Jameson, Analysis and design of numerical schemes for gas dynamics 1, artificial diffusion, upwind biasing, limiters and their effect on multigrid convergence, Int. J. Comput. Fluid Dyn. 4 (1995) 171–218.
[28] J. Li, G. Chen, The generalized Riemann problem method for the shallow water equations with bottom topography, Int. J. Numer. Meth. Eng. 65 (2006) 834–862.
[29] J. Li, Q. Li, K. Xu, Comparisons of the generalized Riemann solver and the gas-kinetic scheme for inviscid compressible flow simulations, J. Comput. Phys. 230 (2011) 5080–5099.
[30] J. Li, T. Zhang, S. Yang, The two-dimensional Riemann problem in gas dynamics, Pitman Monographs and Surveys in Pure and Applied Mathematics, 98, Harlow: Addison Wesley Longman, 1998.
[31] R. Li, T. Tang, P.W. Zhang, Moving mesh methods in multiple dimensions based on harmonic maps, J. Comput. Phys. 170 (2001) 562–588.
[32] R. Li, T. Tang, 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.
[33] X.D. Liu, P.D. Lax, Solution of two-dimensional Riemann problems of gas dynamics by positive schemes, SIAM J. Sci. Comput. 19 (1998) 319–340.
[34] P. Maire, A high-order cell-centered Lagrangian scheme for two–dimensional compressible fluid flows on unstructured meshes, J. Comput. Phys. 228 (2009) 2391–2425.
[35] D.J. Mavriplis, Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes, AIAA-Paper, pp. 2003–3986, June 2003.
[36] D.J. Mavriplis, Unstructured mesh discretizations and solvers for computational aerodynamics, in: 18th AIAA Computational Fluid Dynamics Conference, AIAA 2007–3955, June 2007.
[37] K. Miller, R.N. Miller, Moving finite element, I, SIAM J. Numer. Anal. 18 (1981) 1019–1032.
[38] G. Ni, S. Jiang, K. Xu, Efficient kinetic schemes for steady and unsteady flow simulations on unstructured meshes, J. Comput. Phys. 227 (2008) 3015–3031.
[39] 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.
[40] G.A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, J. Comput. Phys. 27 (1978) 1–31.
[41] H.Z. Tang, T. Tang, Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws, SIAM J. Numer. Anal. 41 (2003) 487–515.

[42] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics: a Practical Introduction, Springer, 1997.

[43] V. Venkatakrishnan, Convergence to steady state solutions of the Euler equations on unstructured grids with limiters, J. Comput. Phys. 118 (1995) 120–130.

[44] D. Wang, X. Wang, A three-dimensional adaptive method based on the iterative grid redistribution, J. Comput. Phys. 199 (2004) 423–436.

[45] A. Winslow, Numerical solution of the quasi–linear Poisson equation, J. Comput. Phys. 1 (1967) 149–172.

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

[47] P.A. Zegeling, W.D. de Boer, H.Z. Tang, Robust and efficient adaptive moving mesh solution of 2-D Euler equation, Contemp. Math. 383 (2005) 375–386.

[48] Z.C. Yang, P. He, H.Z. Tang, A direct Eulerian GRP scheme for relativistic hydrodynamics: one-dimensional case, J. Comput. Phys. 230 (22) (2011) 7964–7987.

[49] Z.C. Yang, H.Z. Tang, A direct Eulerian GRP scheme for relativistic hydrodynamics: two-dimensional case, J. Comput. Phys. 231 (4) (2012) 2116–2139.

[50] R. Zhang, M. Zhang, C.W. Shu, On the order of accuracy and numerical performance of two classes of finite volume WENO schemes, Commun. Comput. Phys. 9 (2011) 807–827.

[51] 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.