A Moving Mesh Method for the Euler Flow Calculations Using a Directional Monitor Function

Huazhong Tang *

Abstract

This paper continues a study of the adaptive grid method proposed recently by Tang and Tang (2003, *SIAM J. Numer. Anal.*, 41, 487-515) for computations of the Euler equations in fluid dynamics. The new feature of the present moving mesh algorithm is the use of a dimensional-splitting type monitor function, which is to increase grid concentration in regions containing shock waves and contact discontinuities or their interactions. Several two–dimensional flow problems are computed to demonstrate the effectiveness of the present adaptive grid algorithm.

Keywords: Euler equations, adaptive grid method, moving mesh method, finite volume method, monitor function.

1 Introduction

In fluid dynamics, the physical solutions usually develop dynamically singular or nearly singular solutions in fairly localized regions, such as shock waves, detonation waves, contact discontinuities, and boundary layers etc. To resolve these large solution variations, the numerical simulations may require extremely fine meshes over a small portion of the physical domain. It becomes very expensive for computations of multi–dimensional problems if uniform mesh is used. It is therefore very necessary to develop an effective, robust, multi-dimensional adaptive grid methods. Successful implementation of the adaptive strategy can increase the accuracy of the numerical approximations and also decrease the computational costs.

*LMAM, School of Mathematical Sciences, Peking University, Beijing 100871, P. R. China. Email: hztang@math.pku.edu.cn

Moving mesh methods are a class of the adaptive grid methods, which have important applications in fluid dynamics. They include the variational grid methods or adaptive grid generation methods, the traditional Lagrange methods, and their variations such as arbitrary Lagrangian–Eulerian methods, the free–Lagrange methods, and the unified coordinate system methods. In the past several decades, there has been important progress in moving mesh methods for partial differential equations, including the variational approach of Winslow [32], Brackbill et al. [3, 4], Dvinsky [9], and Li et al. [16, 17]; moving finite element methods of Millers [21], and Davis and Flaherty[8]; and moving mesh PDEs of Cao et al. [6], Li and Petzold [18], and Ceniceros and Hou [7]. Some recent works on the Lagrange methods can be found in [2, 12, 13, 22] and references therein.

The monitor function is one of the most important issues in the adaptive moving mesh algorithms. The appropriate choice of the monitor will generate grids with good quality in terms of smoothness, skewness, and aspect ratio. Cao et al. [5] gave a general strategy of choosing the monitor function. Some general forms of the monitor function can also be found in [19]. The conventional monitor functions usually depend on magnitude of gradient of the solutions, for example, $\sqrt{1+\alpha|\rho|^2+\beta|\nabla\rho|^2}$. However, the gradient-monitor is not always successful in increasing grid concentration in regions containing shock waves and contact discontinuities or their interactions. Our numerical results will demonstarte this phenomenon. To overcome this drawback, a dimensional–splitting type monitor function will be considered. The moving mesh equations used in this paper is multi–dimensional and more robust than the multidirectional methods, see [19].

In this paper, we are interested in developing moving mesh methods based on a variational approach for the hyperbolic conservation laws including the Euler equations of gas dynamics. Harten and Hyman [11] began the earliest study of the self-adaptive moving mesh methods to improve resolution of shock and contact discontinuity. They applied the Godunov method on a non-uniform mesh where the grids move along the characteristic direction. After their work, many other moving mesh methods in this direction have been proposed in the literature based on combining the variational grid methods with high resolution shock capturing methods. They include works of Azarenok et al. [1], Fazio and LeVeque [10], Liu et al. [20], Saleri and Steinberg [23], Stockie et al. [26], and Tang and Tang [28]. However, it is noticed that many existing moving mesh methods for hyperbolic problems are designed for one space dimension. In 1D, it is generally possible to compute on a very fine grid and so the need for moving mesh methods may not be clear. Multidimensional moving mesh methods are often difficult to be used in fluid dynamics problems since the grid will typically suffer large distortions and possible tangling. It is therefore useful to design a simple and robust moving mesh algorithm for computational problems in fluid dynamics.

Recently, an adaptive moving mesh method for multidimensional hyperbolic conservation laws was proposed by Tang and Tang [28]. The moving mesh algorithm includes two parts: PDE evolution and mesh redistribution. The PDE evolution may be any appropriate high resolution finite volume scheme. The mesh redistribution is an iterative procedure. In each time iteration, grid points are first redistributed by a variational principle, and then the numerical solutions are updated on the resulting new meshes by a *conservative–interpolation* formula.

This paper is to continue the study of [28]. The new feature of the present method is to use a dimensional-splitting type monitor function to increase grid concentration in regions containing shock waves and contact discontinuities or their interactions. This monitor is defined based on an observation of our mesh generation equations.

We will also give numerical comparsions and computations of several two–dimensional flow problems to demonstrate effectiveness of the present moving mesh algorithm in improving resolution of the discontinuities and in increasing grid concentration in the regions containing the discontinuities.

The paper is organized as follows. In Section 2, we present our moving mesh algorithm. Section 2.1 gives the mesh–redistribution strategy based on a variational principle using a new monitor function. Although the new monitor is based on dimensional splitting in the logic space, our moving mesh methods are different from the existing multidirection equidistributions schemes mentioned in [19]. Section 2.2 reviews the derivation of a *conservative–interpolation* formula for solution–updating on new meshes. The PDEs evolution is described in Section 2.3. The detailed solution procedure is outlined in Section 2.4. Section 2.5 gives a simple implementation of the boundary mesh motion in the physical domain, which will be used in our computations. Numerical experiments and comparisons are carried out in Section 3, where several 2D examples are considered. In Section 4, we conclude this work by giving a few remarks.

2 An Adaptive Moving Mesh Method

Consider multi–dimensional hyperbolic conservation laws

$$\frac{\partial U}{\partial t} + \sum_{i=1}^{d} \frac{\partial F_i(U)}{\partial x_i} = H(\mathbf{x}, U), \quad 0 < t \le T, \tag{2.1}$$

subject to the initial data $U(\mathbf{x}, 0) = U_0(\mathbf{x})$, where T is a finite number, d denotes number of spatial dimension, $\mathbf{x} = (x_1, \cdots, x_d)$, U and $F_i(U)$ denote a conservative variable and flux in x_i–direction, respectively, and $H(\mathbf{x}, U)$ is a source term. A typical example of the above equation (2.1) is the Euler equations (3.1).

For convenience, we will restrict our attention to $d = 2$ throughout this paper. Given a partition $\{A_{j+\frac{1}{2},k+\frac{1}{2}}\}$ of the physical domain Ω_p, where $A_{j+\frac{1}{2},k+\frac{1}{2}}$ is a quadrangle with four vertices $\mathbf{x}_{j+p,k+q}$, $0 \le p, q \le 1$, as shown in Fig.1.

The initial data is approximated by a cell average:

$$U^0_{j+\frac{1}{2},k+\frac{1}{2}} = \frac{1}{|A_{j+\frac{1}{2},k+\frac{1}{2}}|} \int_{A_{j+\frac{1}{2},k+\frac{1}{2}}} U_0(\mathbf{x}) \; d\mathbf{x}, \tag{2.2}$$

where $|A_{j+\frac{1}{2},k+\frac{1}{2}}|$ denotes area of the control volume $A_{j+\frac{1}{2},k+\frac{1}{2}}$.

Our adaptive moving mesh methods will contain the following steps: Mesh–redistribution based on variational methods (Section 2.1); conservative solution–updating on new meshes (Section 2.2); and evolution of the governing equations, see Section 2.3. The solution procedure in a detail will be briefly outlined in Section 2.4.

2.1 Mesh–redistribution

The calculus of variations has been proved to provide an excellent opportunity to create new techniques for generation of grids or mesh–redistribution by utilizing the idea of optimization of grid characteristics modeled through appropriate functionals. The grid characteristics include grid smoothes, cell skewness, cell volume, and departure from orthogonality of conformality. We refer the readers to [3, 4, 9, 19, 32] and references therein for details.

In our calculus of variations, the functional is defined by the integral

$$E[\mathbf{x}] = \int_{\Omega_c} \mathcal{G}(\xi, \mathbf{x}, \mathbf{x}_{\xi_i}, \mathbf{x}_{\xi_i \xi_j}) \; d\xi, \tag{2.3}$$

where $\xi = (\xi_1, \cdots, \xi_d)$ is the logical (or computational) coordinates, Ω_c is a logical domain, $\mathbf{x}_{\xi_i}$ and $\mathbf{x}_{\xi_i\xi_j}$ are first– and second–order partial derivatives of $\mathbf{x}$ with respect to ξ_i. An optimal transformation $\mathbf{x} = \mathbf{x}(\xi)$ for the functional (2.3) is the solution of the following system of the Euler–Lagrange equations

$$\mathcal{G}_{x_i} - \frac{\partial}{\partial \xi_j} \mathcal{G}_{\frac{\partial x_i}{\partial \xi_j}} + \frac{\partial^2}{\partial \xi_j \partial \xi_k} \mathcal{G}_{\frac{\partial^2 x_i}{\partial \xi_j \partial \xi_k}} = 0, \qquad i = 1, \cdots, d, \tag{2.4}$$

defined in the interior of the domain Ω_c, where subscripts x_i, $\frac{\partial x_i}{\partial \xi_j}$, and $\frac{\partial^2 x_i}{\partial \xi_j \partial \xi_k}$ are the corresponding partial derivatives of $\mathcal{G}$. Usually, $\mathcal{G}$ can be taken as

$$\mathcal{G} = \sum_{i=1}^{d} (\widetilde{\nabla} x_i)^T G_i \widetilde{\nabla} x_i, \tag{2.5}$$

where $\widetilde{\nabla} = (\partial_{\xi_1}, \cdots, \partial_{\xi_d})^T$, G_i are given symmetric positive definite matrices, called *monitor function*. In general, the monitor functions depend on the underlying solution to be adapted. It is very important to choose suitable monitor function for a moving mesh algorithm in order to increase concentration of grid points in a desired localized region, or improve quality of the adaptive meshes. If $\mathcal{G}$ is of form (2.5), then (2.4) becomes

$$\widetilde{\nabla} \cdot (G_i \widetilde{\nabla} x_i) = 0, \quad i = 1, \cdots, d. \tag{2.6}$$

Solving the Euler–Lagrange equations (2.6) on Ω_c will give directly a coordinate transformation $\mathbf{x} = \mathbf{x}(\xi)$ from the logical domain Ω_c to the physical domain Ω_p. Solving the mesh equation (2.6) is not difficult, since it is defined in the logical domain which in most cases is chosen as a unit square.

In this paper, G_i will be taken as

$$G = \text{diag}\{w_1, \cdots, w_d\}, \tag{2.7}$$

and (2.6) is further reduced to

$$\frac{\partial}{\partial \xi_1}\left(w_1 \frac{\partial x_i}{\partial \xi_1}\right) + \cdots + \frac{\partial}{\partial \xi_d}\left(w_d \frac{\partial x_i}{\partial \xi_d}\right) = 0, \quad i = 1, \cdots, d. \tag{2.8}$$

The control function w_l is traditionally defined by

$$w_l \equiv w = \left(1 + \alpha|\psi|^2 + \beta \sum_{i=1}^{d} \left|\frac{\partial \psi}{\partial \xi_i}\right|^2\right)^{\frac{\gamma}{2}}, \quad 1 \leq l \leq d, \tag{2.9}$$

where α, β and γ are some positive constants, and ψ denotes one of the dependent variables, e.g., $\psi = \rho$, the fluid density.

The monitor function (2.9) means that the function w_l depends on variations of gradient of the underlying solution ψ in all coordinate directions. However, we may find from (2.8) that w_l only plays a role to control movement of a grid point in the ξ_l–direction in the logical space. Therefore, if there is no variation of gradient for the underlying solution in the ξ_l–direction, then it is expected that the grid points should keep almost unchanged in that direction.

Based on this observation, another control function w_l can be defined by

$$w_l = \left(1 + \alpha_l |\psi|^2 + \beta_l \left|\frac{\partial \psi}{\partial \xi_l}\right|^2\right)^{\frac{\gamma_l}{2}}, \quad l = 1, \cdots, d, \tag{2.10}$$

where α_l, β_l and γ_l are some positive constants. It will affect mainly the grid node redistribution in the direction of the logical coordinate lines. Although the monitor $\widehat{G}$ is given based on an idea of dimensional splitting type methods, our moving mesh equations are still different from the existing multidirection equidistributions. The latter is to just simply generate the adaptive meshes by tensor-type extension of the one–dimensional equidistribution algorithm.

Before ending this subsection, we give a finite difference or volume discretization of the mesh equation (2.8). In the remaining of this work, we will concentrate on two-dimensional computations. Divide the logical domain $\Omega_c = \{(\xi_1, \xi_2) = (\xi, \eta) | 0 \le \xi \le 1, 0 \le \eta \le 1\}$ into the square meshes:

$$\left\{(\xi_j, \eta_k) \,\middle|\, \xi_j = j\Delta\xi,\ \eta_k = k\Delta\eta;\ 0 \le j \le J_x + 1, 0 \le k \le J_y + 1\right\},$$

where $\Delta\xi = 1/(J_x + 1)$ and $\Delta\eta = 1/(J_y + 1)$. Correspondingly, the numerical approximations to $\mathbf{x} = \mathbf{x}(\xi, \eta)$ are denoted by $\mathbf{x}_{j,k} = \mathbf{x}(\xi_j, \eta_k)$. The Euler-Lagrange equation (2.6) is discretized by a second–order central difference scheme as

$$\frac{\Delta_-^j \left((w_1)_{j+\frac{1}{2},k} \Delta_+^j (x_i)_{j,k}\right)}{(\Delta\xi)^2} + \frac{\Delta_-^k \left((w_2)_{j,k+\frac{1}{2}} \Delta_+^k (x_i)_{j,k}\right)}{(\Delta\eta)^2} = 0, \tag{2.11}$$

where Δ_+^ν and Δ_-^ν denote forward and backward difference operators respectively in ξ–direction when $\nu = j$ and η–direction when $\nu = k$, and

$$(w_1)_{j+\frac{1}{2},k} = \frac{1}{2}\left((w_1)_{j+\frac{1}{2},k+\frac{1}{2}} + (w_1)_{j+\frac{1}{2},k-\frac{1}{2}}\right),$$
$$(w_2)_{j,k+\frac{1}{2}} = \frac{1}{2}\left((w_2)_{j+\frac{1}{2},k+\frac{1}{2}} + (w_2)_{j-\frac{1}{2},k+\frac{1}{2}}\right).$$

The discrete system (2.11) with Dirichlet boundary condition can be solved by some explicit iteration methods, e.g., Gauss–Seidel iteration, with a fixed number of iterations.

In [28], the monitor function G given in (2.7) with (2.9) is used. The results showed that a unsatisfied resolution was obtained for contact discontinuities. This is the main motivation for suggesting the new monitor (2.7) with (2.10). Moreover, the Gauss–Seidel (G-S) iteration method will be applied in our computations. It is found that G-S iteration is much robust than the Jacobi iteration, even though the latter has a good symmetry.

2.2 Conservative solution interpolation

Let $\mathbf{x}_{j,k}$ and $\widetilde{\mathbf{x}}_{j,k}$ be coordinates of the old and new grid points, respectively, in the sense that the grid point $\mathbf{x}_{j,k}$ will move to position $\widetilde{\mathbf{x}}_{j,k}$ after one G-S iteration for the mesh equations. Use $A_{j+\frac{1}{2},k+\frac{1}{2}}$ and $\widetilde{A}_{j+\frac{1}{2},k+\frac{1}{2}}$ to denote the quadrangles (finite control volumes) with four vertices $(\mathbf{x}_{j+p,k+q})$, and $(\widetilde{\mathbf{x}}_{j+p,k+q})$, $0 \le p, q \le 1$, respectively. They are of similar setup to one shown in Fig.1.

Using a perturbation method, we can derive a *conservative* interpolation scheme to update cell average values $\widetilde{U}_{j+\frac{1}{2},k+\frac{1}{2}}$ over the control volume $\widetilde{A}_{j+\frac{1}{2},k+\frac{1}{2}}$.

Assuming$(\widetilde{x}, \widetilde{y}) = (x - c^x(x,y),\ y - c^y(x,y))$ with small magnitudes of speed (c^x, c^y). We have

$$
\begin{aligned}
\int_{\widetilde{A}_{j+\frac{1}{2},k+\frac{1}{2}}} U(\widetilde{x},\widetilde{y})\ d\widetilde{x}d\widetilde{y} &= \int_{A_{j+\frac{1}{2},k+\frac{1}{2}}} U(x-c^x,\ y-c^y) \det\left(\frac{\partial(\tilde{x},\tilde{y})}{\partial(x,y)}\right) dxdy \\
&\approx \int_{A_{j+\frac{1}{2},k+\frac{1}{2}}} (U(x,y) - c^x U_x - c^y U_y)(1 - c^x_x - c^y_y)\ dxdy \\
&\approx \int_{A_{j+\frac{1}{2},k+\frac{1}{2}}} \Big[U(x,y) - (c^x U)_x - (c^y U)_y\Big]\ dxdy \\
&= \int_{A_{j+\frac{1}{2},k+\frac{1}{2}}} U(x,y)\ dxdy - \Big[(c^2_{\mathbf{n}} U)_{j+1,k+\frac{1}{2}} + (c^4_{\mathbf{n}} U)_{j,k+\frac{1}{2}}\Big] \\
&\qquad - \Big[(c^3_{\mathbf{n}} U)_{j+\frac{1}{2},k+1} + (c^1_{\mathbf{n}} U)_{j+\frac{1}{2},k}\Big],
\end{aligned}
\tag{2.12}
$$

where the higher order terms have been neglected, $c^l_{\mathbf{n}} := c^x n^l_x + c^y n^l_y$ with the normal outward vertor $\mathbf{n}^l = (n_x^l, n_y^l)$ defined as follows:

$$\begin{aligned}
c_{\mathbf{n}}^1 &= \frac{1}{2}(c_{j,k}^x + c_{j+1,k}^x)(y_{j+1,k} - y_{j,k}) - \frac{1}{2}(c_{j,k}^y + c_{j+1,k}^y)(x_{j+1,k} - x_{j,k}),\\
c_{\mathbf{n}}^2 &= \frac{1}{2}(c_{j+1,k}^x + c_{j+1,k+1}^x)(y_{j+1,k+1} - y_{j+1,k}) - \frac{1}{2}(c_{j+1,k}^y + c_{j+1,k+1}^y)(x_{j+1,k+1} - x_{j+1,k}),\\
c_{\mathbf{n}}^3 &= \frac{1}{2}(c_{j+1,k+1}^x + c_{j,k+1}^x)(y_{j,k+1} - y_{j+1,k+1}) - \frac{1}{2}(c_{j+1,k+1}^y + c_{j,k+1}^y)(x_{j,k+1} - x_{j+1,k+1}),\\
c_{\mathbf{n}}^4 &= \frac{1}{2}(c_{j,k+1}^x + c_{j,k}^x)(y_{j,k} - y_{j,k+1}) - \frac{1}{2}(c_{j,k+1}^y + c_{j,k}^y)(x_{j,k} - x_{j,k+1}),
\end{aligned}$$

and $(c_{\mathbf{n}}U)_{j+p,k+\frac{1}{2}}$ and $(c_{\mathbf{n}}U)_{j+\frac{1}{2},k+q}$, $p,q = 0$ or 1, denote the values of $c_{\mathbf{n}}U$ on the corresponding surface of the control volume $A_{j+\frac{1}{2},k+\frac{1}{2}}$. They will be approximated by using an upwind scheme. For example, the term $(c_{\mathbf{n}}U)_{j+1,k+\frac{1}{2}}$ may be approximated by

$$(c_{\mathbf{n}}U)_{j+1,k+\frac{1}{2}} = \frac{c_{\mathbf{n}}^2}{2}(U_{j+\frac{3}{2},k+\frac{1}{2}} + U_{j+\frac{1}{2},k+\frac{1}{2}}) - \frac{|c_{\mathbf{n}}^2|}{2}(U_{j+\frac{3}{2},k+\frac{1}{2}} - U_{j+\frac{1}{2},k+\frac{1}{2}}). \tag{2.13}$$

The above approximation is only first order accurate in space. In order to avoid large numerical dissipation, we will use the reconstruction technique [31], which is to be described in Section 2.3.

From (2.12), a conservative-interpolation formula is obtained:

$$\begin{aligned}
&|\widetilde{A}_{j+\frac{1}{2},k+\frac{1}{2}}|\widetilde{U}_{j+\frac{1}{2},k+\frac{1}{2}} = |A_{j+\frac{1}{2},k+\frac{1}{2}}|U_{j+\frac{1}{2},k+\frac{1}{2}}\\
&\quad-\Big[(c_{\mathbf{n}}^2U)_{j+1,k+\frac{1}{2}} + (c_{\mathbf{n}}^4U)_{j,k+\frac{1}{2}}\Big] - \Big[(c_{\mathbf{n}}^3U)_{j+\frac{1}{2},k+1} + (c_{\mathbf{n}}^1U)_{j+\frac{1}{2},k}\Big],
\end{aligned} \tag{2.14}$$

where $|\widetilde{A}|$ and $|A|$ denote the areas of the control volumes $\widetilde{A}$ and A, respectively. It can be verified that the above interpolation scheme satisfies the following mass–conservation property:

$$\sum_{j,k} |\widetilde{A}_{j+\frac{1}{2},k+\frac{1}{2}}|\widetilde{U}_{j+\frac{1}{2},k+\frac{1}{2}} = \sum_{j,k} |A_{j+\frac{1}{2},k+\frac{1}{2}}|U_{j+\frac{1}{2},k+\frac{1}{2}}. \tag{2.15}$$

It is also possible to derive a conservative-interpolation from geometric point of view. For one-dimensional case, we have

$$\int_{\widetilde{x}_j}^{\widetilde{x}_{j+1}} U(x)\ dx = \int_{x_j}^{x_{j+1}} U(x)\ dx + \left(\int_{\widetilde{x}_j}^{x_j} U(x)\ dx - \int_{\widetilde{x}_{j+1}}^{x_{j+1}} U(x)\ dx\right),$$

i.e.

$$(\widetilde{x}_{j+1} - \widetilde{x}_j)\widetilde{U}_{j+\frac{1}{2}} = (x_{j+1} - x_j)U_{j+\frac{1}{2}} + \left(\int_{\widetilde{x}_j}^{x_j} U(x)\ dx - \int_{\widetilde{x}_{j+1}}^{x_{j+1}} U(x)\ dx\right).$$

According to the the locations of the new points $\widetilde{x}_j$ and $\widetilde{x}_{j+1}$, we can give a suitable approximation for the last two integrals, which leads to the conservative-interpolation formula as proposed in [28]. However, it will become very complicated to construct multidimensional conservative-interpolation using this geometrical approach, although we believe that multidimensional conservative-interpolation obtained by using geometrical approach are more accurate and more suitable for computations of the complex problems.

2.3 PDE evolution

Assume that a partition $\mathbf{x}_{j,k} = (x_{j,k}, y_{j,k})$ of the physical domain Ω_p is given, and the initial data $U^0_{j+\frac{1}{2},k+\frac{1}{2}}$ is defined by (2.2). Integrating (2.1) over the finite control volume $A_{j+\frac{1}{2},k+\frac{1}{2}}$ gives

$$\frac{\partial}{\partial t}\iint_{A_{j+\frac{1}{2},k+\frac{1}{2}}} U\ dxdy + \sum_{l=1}^{4} F_{\mathbf{n}^l}(U)|_{(x,y)\in s_i} = \iint_{A_{j+\frac{1}{2},k+\frac{1}{2}}} H\ dxdy, \tag{2.16}$$

where s_i, $i = 1, 2, 3, 4$, are four sides of $A_{j+\frac{1}{2},k+\frac{1}{2}}$, $F_{\mathbf{n}^l} = \mathbf{F}\cdot\mathbf{n}^l$, and $\mathbf{n}^l = (n_x^l, n_y^l)$ are the normal outward vertors, defined as

$$\begin{aligned}
&\mathbf{n}^1 = (y_{j+1,k} - y_{j,k}, x_{j,k} - x_{j+1,k}), \quad \mathbf{n}^2 = (y_{j+1,k+1} - y_{j+1,k}, x_{j+1,k} - x_{j+1,k+1}),\\
&\mathbf{n}^3 = (y_{j,k+1} - y_{j+1,k+1}, x_{j+1,k+1} - x_{j,k+1}), \quad \mathbf{n}^4 = (y_{j,k} - y_{j,k+1}, x_{j,k+1} - x_{j,k}).
\end{aligned}$$

By assuming $F_{\mathbf{n}^l} = F^+_{\mathbf{n}^l} + F^-_{\mathbf{n}^l}$, we obtain a first order finite volume scheme for (2.1):

$$\begin{aligned}
U^{n+1}_{j+\frac{1}{2},k+\frac{1}{2}} =& U^n_{j+\frac{1}{2},k+\frac{1}{2}} - \frac{\Delta t}{|A_{j+\frac{1}{2},k+\frac{1}{2}}|}\Big(\sum_{l=1}^{4} F^+_{\mathbf{n}^l}(U^n_{j+\frac{1}{2},k+\frac{1}{2}}) + F^-_{\mathbf{n}^1}(U^n_{j+\frac{1}{2},k-\frac{1}{2}})\\
&+ F^-_{\mathbf{n}^2}(U^n_{j+\frac{3}{2},k+\frac{1}{2}}) + F^-_{\mathbf{n}^3}(U^n_{j+\frac{1}{2},k+\frac{3}{2}}) + F^-_{\mathbf{n}^4}(U^n_{j-\frac{1}{2},k+\frac{1}{2}})\Big)\\
&+ \Delta t H^n_{j+\frac{1}{2},k+\frac{1}{2}}.
\end{aligned} \tag{2.17}$$

The spatial accuracy of the scheme (2.17) can be improved by using the reconstruction technique. Following [31], we use the piecewise linear function

$$\overline{U}^n_{j+\frac{1}{2},k+\frac{1}{2}}(x,y) = U^n_{j+\frac{1}{2},k+\frac{1}{2}} + \mathbf{S}^n_{j+\frac{1}{2},k+\frac{1}{2}}\cdot(\mathbf{x} - \mathbf{x}_{j+\frac{1}{2},k+\frac{1}{2}}),\ (x,y)\in A_{j+\frac{1}{2},k+\frac{1}{2}}, \tag{2.18}$$

to replace the piecewise constant function $U^n_{j+\frac{1}{2},k+\frac{1}{2}}$ at each time level t_n, where $\mathbf{S}^n_{j+\frac{1}{2},k+\frac{1}{2}}$

is an approximation of the gradient ∇U. This gives a second order scheme:

$$
\begin{aligned}
U^{n+1}_{j+\frac{1}{2},k+\frac{1}{2}} =& U^n_{j+\frac{1}{2},k+\frac{1}{2}} - \frac{\Delta t}{|A_{j+\frac{1}{2},k+\frac{1}{2}}|}\Big(\sum_{l=1}^{4} F^+_{\mathbf{n}^l}(\overline{U}^n_{j+\frac{1}{2},k+\frac{1}{2}}(\mathbf{x}_{c^l})) + F^-_{\mathbf{n}^1}(\overline{U}^n_{j+\frac{1}{2},k-\frac{1}{2}}(\mathbf{x}_{c^1})) \\
&+ F^-_{\mathbf{n}^2}(\overline{U}^n_{j+\frac{3}{2},k+\frac{1}{2}}(\mathbf{x}_{c^2})) + F^-_{\mathbf{n}^3}(\overline{U}^n_{j+\frac{1}{2},k+\frac{3}{2}}(\mathbf{x}_{c^3})) + F^-_{\mathbf{n}^4}(\overline{U}^n_{j-\frac{1}{2},k+\frac{1}{2}}(\mathbf{x}_{c^4}))\Big) \\
&+ \Delta t H^n_{j+\frac{1}{2},k+\frac{1}{2}} =: U^n_{j+\frac{1}{2},k+\frac{1}{2}} + \Delta t L_{j+\frac{1}{2},k+\frac{1}{2}}(U^n),
\end{aligned} \tag{2.19}
$$

where $\mathbf{x}_{c^l}$ denotes the coordinates of the middle of the lth edge of the control element $A_{j+\frac{1}{2},k+\frac{1}{2}}$.

In our computations, the van Leer's slope limiter is used to suppress appearance of the numerical oscillation. Moreover, the 2rd order TVD Runge-Kutta time discretization [25]

$$
U^*_{j+\frac{1}{2},k+\frac{1}{2}} = U^n_{j+\frac{1}{2},k+\frac{1}{2}} + \Delta t L_{j+\frac{1}{2},k+\frac{1}{2}}(U^n), \tag{2.20}
$$

$$
U^{n+1}_{j+\frac{1}{2},k+\frac{1}{2}} = \frac{1}{2}\big(U^n_{j+\frac{1}{2},k+\frac{1}{2}} + U^*_{j+\frac{1}{2},k+\frac{1}{2}} + \Delta t L_{j+\frac{1}{2},k+\frac{1}{2}}(U^*)\big), \tag{2.21}
$$

is used to replace the forward Euler time discretization (2.19).

Remark 2.1. *In practice, we may simplify the problem in question by applying the initial reconstruction technique along each normal direction of the finite control volume.*

2.4 Solution procedure

The solution procedure of our adaptive mesh strategy for two-dimensional hyperbolic problems can be summarized as follows:

Step i Give an initial partition $\mathbf{x}^{[0]}_{j,k} = \big(x^{[0]}_{j,k}, y^{[0]}_{j,k}\big) := (x_{j,k}, y_{j,k})$ of the physical domain Ω_p and a uniform (fixed) partition of the logical domain Ω_c, and compute grid values $U^{[0]}_{j+\frac{1}{2},k+\frac{1}{2}}$ by cell average over control volume $A^{[0]}_{j+\frac{1}{2},k+\frac{1}{2}}$ based on the initial data $U(\mathbf{x}, 0)$.

Step ii For $\nu = 0, 1, 2, \ldots, M$, with M a fixed number, do the following:

(a). Move grid points $\mathbf{x}^{[\nu]}_{j,k}$ to $\mathbf{x}^{[\nu+1]}_{j,k}$ by solving (2.11) with one Gauss–Seidel iteration.

(b). Compute $\{U^{[\nu+1]}_{j+\frac{1}{2},k+\frac{1}{2}}\}$ over the new control volume $A^{[\nu+1]}_{j+\frac{1}{2},k+\frac{1}{2}}$ using the conservative-interpolation (2.14).

Step iii Evolve the underlying PDEs by using 2D high-resolution finite volume methods given in Section 2.3 on the mesh $\mathbf{x}_{j,k}^{[M+1]}$ to obtain the numerical approximations $U_{j+\frac{1}{2},k+\frac{1}{2}}^{n+1}$ at the time level t_{n+1}.

Step iv If $t_{n+1} < T$, then let $U_{j+\frac{1}{2},k+\frac{1}{2}}^{[0]} := U_{j+\frac{1}{2},k+\frac{1}{2}}^{n+1}$ and $\mathbf{x}_{j,k}^{[0]} := \mathbf{x}_{j,k}^{[M+1]}$, and go to Step ii.

2.5 Boundary redistribution

In practical flow situations, the discontinuities may initially exist in boundaries or move to boundaries in a later time. As a consequence, movement of boundary points should be made in order to improve the quality of the adaptive solutions near boundary. A simple redistribution strategy is proposed as follows. For convenience, our attention is restricted to the case that the physical domain Ω_p is rectangular. Assume a new set of grid points $\tilde{\mathbf{x}}_{j,k}$ is obtained in Ω_p by solving the moving mesh equation (2.11). Then the speeds of internal grid points $\mathbf{x}_{j,k}$ are given by:

$$(c^x, c^y)_{j,k} := (x^{[\nu]} - x^{[\nu+1]}, y^{[\nu]} - y^{[\nu+1]})_{j,k}, \quad \text{for} \quad 1 \le j \le J_x,\ 1 \le k \le J_y.$$

We assume that the points on the boundaries are moving with the same speed as the tangential component of the speed for the internal points adjacent to those boundary points, namely

$$\begin{aligned}
(c^x, c^y)_{0,k} &= (0, c^y_{1,k}),\ (c^x, c^y)_{J_x+1,k} = (0, c^y_{J_x,k}) \qquad && 1 \le k \le J_y, \\
(c^x, c^y)_{j,0} &= (c^x_{j,1}, 0),\ (c^x, c^y)_{j,J_y+1} = (c^x_{j,J_y}, 0), \qquad && 1 \le j \le J_x.
\end{aligned}$$

Thus the new boundary points $\widetilde{\mathbf{x}}_{0,k}$, $\widetilde{\mathbf{x}}_{J_x+1,k}$, $\widetilde{\mathbf{x}}_{j,0}$ and $\widetilde{\mathbf{x}}_{j,J_y+1}$ are defined by

$$\begin{aligned}
(\widetilde{x}, \widetilde{y})_{j,k} &= (x, y)_{j,k} - (c^x, c^y)_{j,k}, \quad j = 0 \text{ or } J_x + 1, \quad 1 \le k \le J_y, \\
(\widetilde{x}, \widetilde{y})_{j,k} &= (x, y)_{j,k} - (c^x, c^y)_{j,k}, \quad 1 \le j \le J_x, \quad k = 0 \text{ or } J_y + 1.
\end{aligned}$$

Numerical experiments show that the above procedure to move the boundary points is useful in improving the solution resolution.

If the physical domain Ω_p is not a rectangular, then we introduce an intermediate or parameter space $\mathcal{D}$ with Cartesian coordinates $\zeta = (\zeta_1, \zeta_2)$ where $\mathcal{D}$ is taken as the unit square. At $t = 0$, we construct an initial map $\mathbf{x} = \mathbf{x}(\zeta) : \mathcal{D} \to \Omega_p$ by calculus of variations or using algebric grid generation method, if there does not exist any map between Ω_p and

$\mathcal{D}$ with an explicit algebraic expression. For $t > 0$, we first move the grid points in $\mathcal{D}$ according to the previous algorithm, i.e. to solve the following moving mesh equations

$$\frac{\Delta_-^j\left((w_1)_{j+\frac{1}{2},k}\Delta_+^j(\zeta_i)_{j,k}\right)}{(\Delta\xi)^2}+\frac{\Delta_-^k\left((w_2)_{j,k+\frac{1}{2}}\Delta_+^k(\zeta_i)_{j,k}\right)}{(\Delta\eta)^2}=0,\ i=1,2. \tag{2.22}$$

Next we update the underlying solutions by using the conservative-interpolation formula given in Section 2.2, and the Cartesian coordinates $\mathbf{x}$ in Ω_p by a non-conservative high resolution interpolation used in [30], i.e, using a high resolution scheme to solve the following discrete equation:

$$\mathbf{x}(\widetilde{\zeta}) = \mathbf{x}(\zeta) + (\widetilde{\zeta} - \zeta)\cdot\nabla_\zeta\mathbf{x}. \tag{2.23}$$

Here $\mathbf{x}$ is considered as the functions of ζ, and $\mathbf{x}_{j,k}$ is the nodal discretization. Using a non-conservation interpolation instead of the mesh generation procedure may reduce the computational cost due to numerical grid generation.

The idea of introducing an intermediate space has also been widely used in numerical grid generations, see e.g., [19, 27] for details.

3 Numerical Experiment

In this section, we solve the Euler equations governing two–dimensional inviscid compressible fluid flows

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ E\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ u(E+p)\end{pmatrix}+\frac{\partial}{\partial y}\begin{pmatrix}\rho v\\ \rho vu\\ \rho v^2+p\\ v(E+p)\end{pmatrix}=0, \tag{3.1}$$

where ρ is the density, (u, v) is the velocity vector, E is the total energy, p is the pressure. Assume that the pressure is related to the total energy by $E = \frac{p}{\gamma-1} + \frac{1}{2}\rho(u^2+v^2)$ with $\gamma = 1.4$. The physical fluxes are approximated by the kinetic flux–vector splitting method. For example, the flux $(F_1)_{j,k+\frac{1}{2}}$ in x–direction can be approximated by

$$\begin{aligned}(F_1)_{j,k+\frac{1}{2}} &= (F_1)^+_{j-\frac{1}{2},k+\frac{1}{2}} + (F_1)^-_{j+\frac{1}{2},k+\frac{1}{2}},\\ (F_1)^\pm &= \left(\rho[\tilde{u}^1]_\pm, \rho[\tilde{u}^2]_\pm, \rho[\tilde{u}]_\pm[\tilde{v}^1], \frac{1}{2}\left([\tilde{u}^3]_\pm + [\tilde{u}]_\pm[\xi^2]\right)\right)^T,\end{aligned}$$

where

$$
\begin{aligned}
&[\tilde{v}^1] = v, \quad [\xi^2] = \frac{K}{2\lambda},\\
&[\tilde{u}^0]_\pm = \frac{1}{2}\mathrm{erfc}(\mp\sqrt{\lambda}u),\\
&[\tilde{u}^1]_\pm = u[\tilde{u}^0]_\pm \pm \frac{e^{-\lambda u^2}}{\sqrt{\pi\lambda}},\\
&[\tilde{u}^{n+2}]_\pm = u[\tilde{u}^{n+1}]_\pm + \frac{n+1}{2\lambda}[\tilde{u}^n]_\pm, \ n \geq 0.
\end{aligned}
$$

Here $\lambda = \rho/2p$, $K = -2+2/(\gamma-1)$ and $\mathrm{erfc}(x)$ denotes the complementary error function.

In the following, we will apply the moving mesh algorithm described in Section 2.4 to our numerical experiments, and compare the monitor function G defined in (2.7) with (2.10) to the monitor function G with (2.9). Unless otherwise specified, the number of G-S iteration for the mesh equations is taken as 5; $\alpha_l = \alpha = 0$. Throughtout the computations, the CFL number used is 0.24. The computations are run on PC Pentium–III with 800 MHz.

Example 3.1. *2D Riemann problem I: shock waves [15, 24].*

The initial data is chosen as

$$
W(x,y,0) = \begin{cases}
(1.1,\ 0.0,\ 0.0,\ 1.1) & \text{if } x > 0.5, \quad y > 0.5,\\
(0.5065,\ 0.8939,\ 0.0,\ 0.35) & \text{if } x < 0.5, \quad y > 0.5,\\
(1.1,\ 0.8939,\ 0.8939,\ 1.1) & \text{if } x < 0.5, \quad y < 0.5,\\
(0.5065,\ 0.0,\ 0.8939,\ 0.35) & \text{if } x > 0.5, \quad y < 0.5,
\end{cases}
$$

which corresponds to the case of left forward shock, right backward shock, upper backward shock, and lower forward shock, $W = (\rho, u, v, p)^T$. We refer the readers to [15, 24] for details.

Fig. 2 shows the adaptive meshes with 50×50 grid points at $t = 0.25$ obtained by using two monitor functions (2.10) and (2.9) with $\beta_l = \beta = 8$ and $\gamma_l = \gamma = 1$. It is observed that using the monitor function $\widehat{G}$ can increase grid concentration in shock curves and in particular near two shock interaction points (0.92,0.3) and (0.3,0.92). Figs. 3 and 4 show the adaptive meshes and fluid density at $t = 0.25$ with 50×50 grid points, obtained by $\widehat{G}$ and G with $\beta_l = \beta = 8$ and $\gamma_l = \gamma = \frac{3}{2}$, respectively. A sharper shock resolution and a better grid concentration are obtained by using the monitor function $\widehat{G}$. The results in

Fig. 3 have almost the same resolution as ones given in [28] with the monitor function G and 100×100 grid points.

Example 3.2. *2D Riemann problem II: contact discontinuities [15, 24].*

The initial data is chosen as

$$W(x,y,0) = \begin{cases} (1,\ 0.75,\ -0.5,\ 1) & \text{if } x > 0.5, \quad y > 0.5, \\ (2,\ 0.75,\ 0.5,\ 1) & \text{if } x < 0.5, \quad y > 0.5, \\ (1,\ -0.75,\ 0.5,\ 1) & \text{if } x < 0.5, \quad y < 0.5, \\ (3,\ -0.75,\ -0.5,\ 1) & \text{if } x > 0.5, \quad y < 0.5, \end{cases}$$

which corresponds to the case of left forward contact, right backward contact, upper backward contact, and lower forward contact. The purpose of computing this problem is to test the ability of our adaptive moving mesh algorithm in resolving contact discontinuities. Fig. 5 shows the adaptive meshes with 50×50 grid points at $t = 0.3$ obtained by using the two different monitor functions $\widehat{G}$ and G with $\beta_l = \beta = 8$ and $\gamma_l = \gamma = 1$. It is seen from these results that the monitor $widehatG$ produces better mesh quality in terms of smoothness, skewness, and aspect ratio. The adaptive meshes and contour of the density in Fig. 6 are obtained by using 100×100 grid points and the monitor $\widehat{G}$. They have a better resolution than ones given in [28] where the monitor G is used. It also shows that the monitor $\widehat{G}$ is successful in increasing grid concentration near contact discontinuity.

Example 3.3. *2D spherical Riemann problem [14].*

This example is a spherical Riemann problem between two parallel walls. The main feature of the solution is the complex interactions among the waves and between waves and two walls. The problem is cylindrically symmetric within our considered time interval [0,0.7]. Hence we may solve 2D axisymmetric Euler equations, and use x to correspond to the radial coordinate. Moreover, a source term $H = -\frac{1}{x}[\rho u, \rho u^2, \rho uv, u(E+p)]^T$ is added at the right hand side of the Euler equations (3.1).

Initially the air is at rest with the density $\rho = 1$, and the pressure $p = 1$ everywhere except in a sphere centered at $(0, 0.4)$ with radius 0.2. Inside the sphere, the pressure $p = 5$. The initial jump in pressure results in a strong outward moving shock wave and contact discontinuity and an inward rarefaction wave. This inward moving wave causes further a local "implosion", and create a second outward moving shock wave. The computational domain is taken as $[0, 1.4] \times [0, 1]$, and the two walls are located at $y = 0$ and $y = 1$. The symmetric boundary condition is used at $x = 0$, and free outflow is specified at $x = 1.4$. The reflection conditions are used at two walls.

Figs. 7–9 show the adaptive meshes and the pressure distributions at $t = 0.7$ by using $\widehat{G}$ and G with $\beta_l = \beta = 10$ and $\gamma_l = \gamma = 1$, respectively. A 125×100 mesh is used for Figs. 7 and 8 and a 250×200 mesh for Fig. 9. The flow at $t = 0.7$ consists of several shocks and strong contact/tangential discontinuities surrounding the low density near the center od the domain. From these figures, it is observed that some main wave structures are computed with sharper resolutions. Our results in Fig. 9 can be compared with one given in [14] wich is obtained using a 600×400 mesh (Fig. 6d there). It is seen from Figs. 7 and 8 that the adaptation effect is more satisfactory with the use of the monitor $\widehat{G}$. The mesh is smoother in the region $[1.2, 1.4] \times [0, 1]$. With a fine mesh of 250×200 grid points, the problem is well resolved, as seen in Fig. 9.

Example 3.4. *The double-Mach reflection problem [33].*

This problem was studied extensively by Woodward and Colella [33] and later by many others. We use exactly the same setup as in [33], i.e. the same initial and boundary conditions and same solution domain $\Omega_p = [0, 4] \times [0, 1]$. Initially a right-moving shock with Mach 10 is positioned at point $(x, y) = (\frac{1}{6}, 0)$ and makes a 60° angle with the x-axis. More precisely, the initial data are

$$U(x, y, 0) = \begin{cases} U_L, & \text{for} \quad y \geq h(x, 0), \\ U_R, & \text{otherwise}, \end{cases}$$

where $U = [\rho, \rho u, \rho v, E]^T$, the left state U_L, the right state U_R, and the shock location are

$$\begin{aligned} &U_L = (8,\ 57.1597,\ -33.0012,\ 563.544)^T, \\ &U_R = (1.4,\ 0.0\ 0.0,\ 2.5)^T, \\ &h(x, t) = \sqrt{3}(x - 1/6) - 20t. \end{aligned}$$

Figs. 10 and 11 show the adaptive meshes and contours of the density at $t = 0.2$ with 160×80 grid points within a cut domain $[0, 3.1] \times [0, 1]$. From these figures, the same phenomenon as in the previous examples are observed, although it is less obvious. Within the local regions near points (2.7,0.41) and (2.6, 0.0), respectively, the grid concentration obtained by $\widehat{G}$ is better than the one given by G. The high deformed cells have been produced by G in some local regions. Moreover, the local orthogonality of the meshes in Fig. 10 is better than one Fig. 11.

Example 3.5. *Supersonic flows around a double ellipse configuration.*

We consider is a steady problem on supersonic flows around a double ellipse configuration, which is to show numerical implementation of the present adaptive moving grid algorithm for solving the problems in a complex domain. The double ellipse configuration is defined by

$$\left(\frac{x}{0.06}\right)^2 + \left(\frac{y}{0.015}\right)^2 = 1,$$
$$\left(\frac{x}{0.035}\right)^2 + \left(\frac{y}{0.025}\right)^2 = 1,$$

for $x \leq 0$, and $y = -0.015$ (or 0.025) in the lower (or upper) plane when $0 \leq x \leq 0.016$.

The problem is initialized by a freestream with a Mach number 3 and the angle of attack $20°$. A reflection boundary condition is imposed at the surface of the double ellipse configuration, i.e.$k = 1$, an inflow boundary condition is applied at the far boundary condition, i.e. $k = J_y$, and an outflow boundary condition is applied at $x = 0.016$ ($j = 1$ and J_x). In this example, the monitor function $\widehat{G}$ is used with $\beta_l = 0.4$ and $\alpha_l = \gamma_l = 1$. Since the physical domain is non-convex, we need to introduce an intermediate space $\mathcal{D}$ as one described in Section 2.5. The computational meshes in Ω_p and the pressure contours are shown in Figs. 12 and 13. The comparison of the results shows a significant improvement of the shock resolution.

4 Concluding Remarks

In this paper, we have further studied the adaptive moving mesh method proposed in [28] for computations of the Euler equations in fluid dynamics. Our adaptive mesh algorithm includes two parts: PDE evolution and mesh redistribution. The first part is an appropriate high resolution finite volume scheme, and the second part is an iterative procedure including redistribution of grid points and a conservative solution–interpolation. Because the two parts are decoupled, the present algorithm may be extended to three-dimensional Euler flow calculations following Tang and Tang [29].

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 magnitude of 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. To improve the performance of the adaptive method, a directional monitor function has been proposed in this work. Although the monitor function is defined based on dimension-splitting idea, it is different from the existing multidirection-equidistribution approachs. The latter is just a direct tensor-product extension of the 1D equidistribution strategy equidistribution scheme.

We have also presented some numerical computations for several test problems and numerical comparisons. The results show that the new monitor function is more effective than the monitor used in [28], which improves the resolution of the discontinuities, by clustering more grid points within the regions containing shocks and contact discontinuities.

Acknowledgment

This research is partially supported by the Special Funds for Major State Basic Research Projects of China, the National Natural Science Foundation of China, and the Alexander von Humboldt foundation and IAN of Otto–von–Guericke Universität Magdeburg. The author also wishes to thank Prof. Tao Tang for numerous discussions during the preparation of this work.

References

[1] B.N. Azarenok, S. A. Ivanenko and T. Tang, Adaptive mesh redistribution method based on Godunov's scheme, *Commun. Math. Sci.*, 1(2003), pp. 152-179.

[2] G.J. Ball, A free-Lagrange method for unsteady compressible flow: Simulation of a confined cylindrical blast wave, *Shock Waves*, 5(1996), pp. 311–325.

[3] J.U. Brackbill, An adaptive grid with directional control, *J. Comput. Phys.*, 108(1993), pp. 38-50.

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

[5] W.M. Cao, W.Z. Huang, and R.D. Russell, A study of monitor functions for two–dimensional adaptive mesh generation, *SIAM J. Sci. Comput.*, 20(1999), pp. 1978–1999.

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

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

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

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

[10] R. Fazio and R. LeVeque, Moving–mesh methods for one–dimensional hyperbolic problems using CLAWPACK, *Comp. Math. Appl.*, 45(2003), pp. 273-298.

[11] A. Harten and J.M. Hyman, Self–adjusting grid methods for one–dimensional hyperbolic conservation laws, *J. Comput. Phys.*, 50(1983), pp. 235–269.

[12] C.W. Hirt, A.A. Amsden, and J.L. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, *J. Comput. Phys.* 14(1974), pp. 227-253; 135(1997), pp. 203-216.

[13] W. H. Hui, P. Y. Li and Z. W. Li, A unified coordinate system for solving the two-dimensional Euler equations, *J. Comput. Phys.*, 153(1999), pp. 596-637.

[14] J.O. Langseth and R.J. LeVeque, A wave propagation method for three–dimensional hyperbolic conservation laws, *J. Comput. Phys.*, 165(2000), pp. 126–166.

[15] P.D. Lax and X.D. Liu, Solutions of two–dimensional Riemann problems of gas dynamics by positive schemes, *SIAM J. Sci. Comput.*, 19(1998), pp. 319–340.

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

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

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

[19] V.D. Liseikin, Grid Generation Methods, Springer, 1999.

[20] F. Liu, S. Ji and G. Liao, An adaptive grid method and its application to steady Euler flow calculations, *SIAM J. Sci. Comput.*, 20(1998), pp. 811-825.

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

[22] R.B. Pember and R.W. Anderson, A comparison of staggered-mesh Lagrange plus remap and cell-centered direct Eulerian Godunov schemes for Eulerian shock hydrodynamics, Lawrence Livermore National Laboratory, CA, UCRL-JC-139820, 1998.

[23] K. Saleri and S. Steinberg, Flux-corrected transport in a moving grid, *J. Comput. Phys.*, 111(1994), pp. 24-32.

[24] C.W. Schulz–Rinne, J.P. Collins, and H.M. Glaz, *Numerical solution of the Riemann problem for two–dimensional gas dynamics*, *SIAM J. Sci. Comput.*, 14(1993), pp. 1394–1414.

[25] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, *J. Comput. Phys.* 77(1988), pp. 439-471.

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

[27] J.F. Thompson, B.K. Soni, and N.P. Weatherill, Handbook of Grid Generation, CRC Press, Boca Raton, London, New York, Washington, D.C. 1999.

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

[29] H.Z. Tang and T. Tang, Multi–dimensional moving mesh methods for shock computations, to appear in Special Volume of Contemporary Mathematics, AMS, edited by S.Y. Cheng, C.W. Shu, and T. Tang, 2003.

[30] H.Z. Tang, T. Tang, and P.W. Zhang, An adaptive mesh redistribution method for nonlinear Hamilton-Jacobi equations in two- and three–dimensions, to appear in *J. Comput. Phys.*, 2003. Also available at `http://www.math.hkbu.edu.hk/~ttang`, 2001.

[31] B. van Leer, Towards the ultimate conservative difference scheme, V. A second order sequel to Godunov's method, *J. Comput. Phys.*, 32(1979), pp. 101–136.

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

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

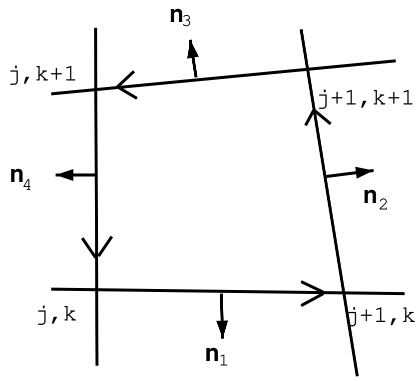

Figure 1: An 2D finite control volume $A_{j+\frac{1}{2},k+\frac{1}{2}}$.

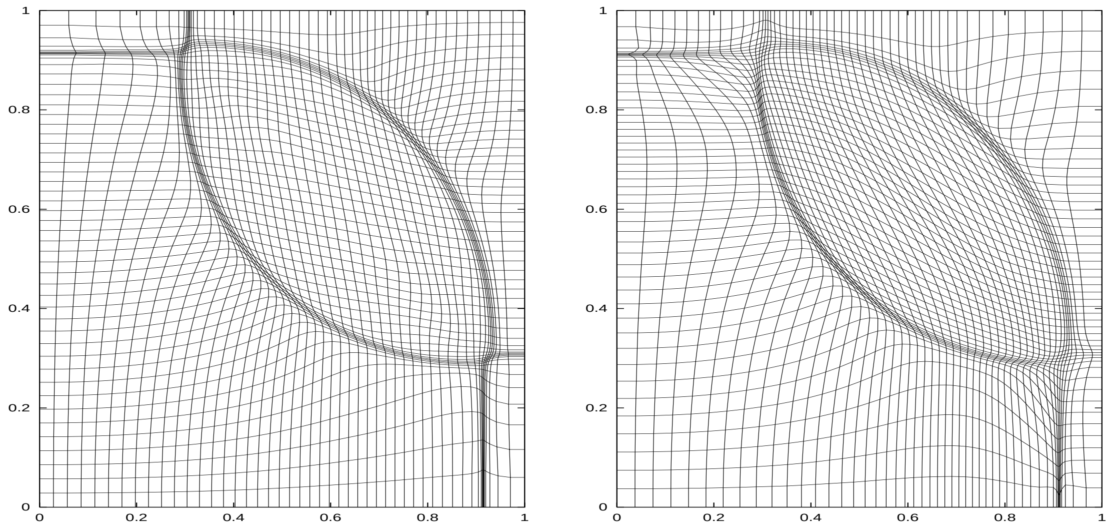

Figure 2: Example 3.1. The adaptive mesh at $t = 0.25$. Left: $\widehat{G}$ with $\beta_l = 8$ and $\gamma_l = 1$; Right: G with $\beta = 8$ and $\gamma = 1$.

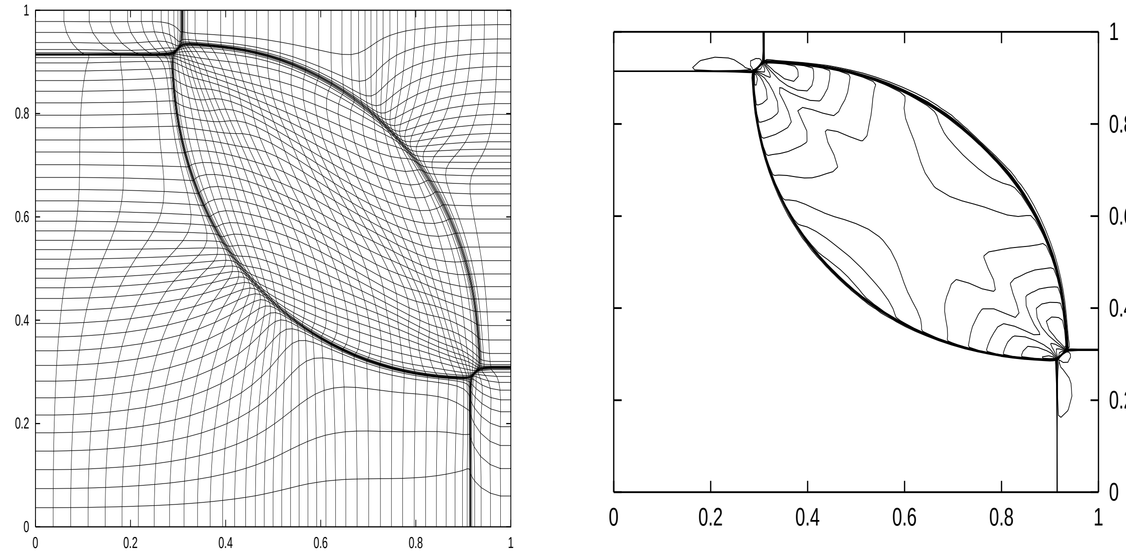

Figure 3: Example 3.1. The adaptive solutions at $t = 0.25$ with 50×50 grid points, obtained by using the monitor $\widehat{G}$ with $\beta_l = 8$ and $\gamma_l = \frac{3}{2}$. Left: adaptive mesh; Right: the density (30 equally contour lines).

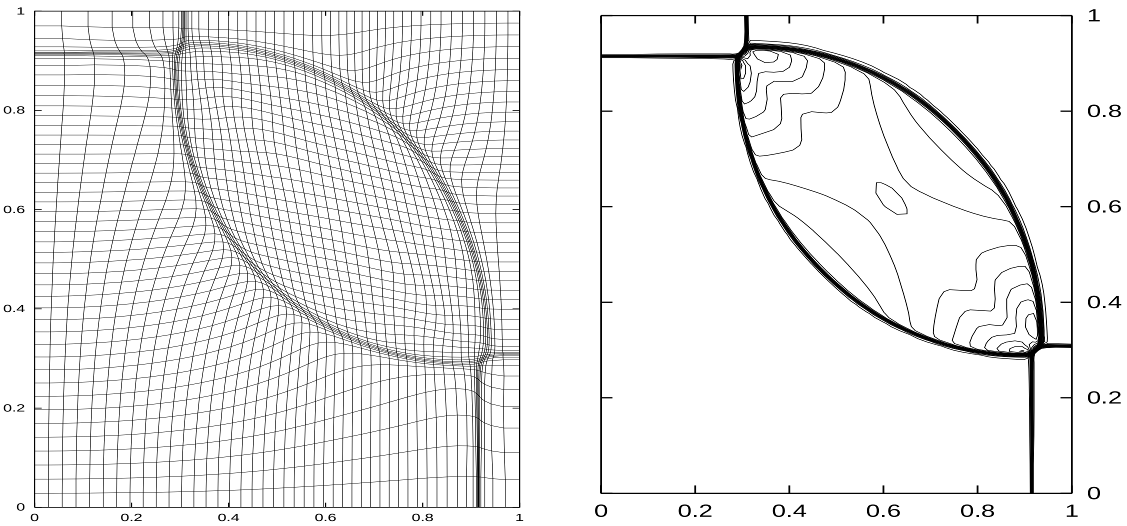

Figure 4: Same as Fig.3, except for the monitor function G.

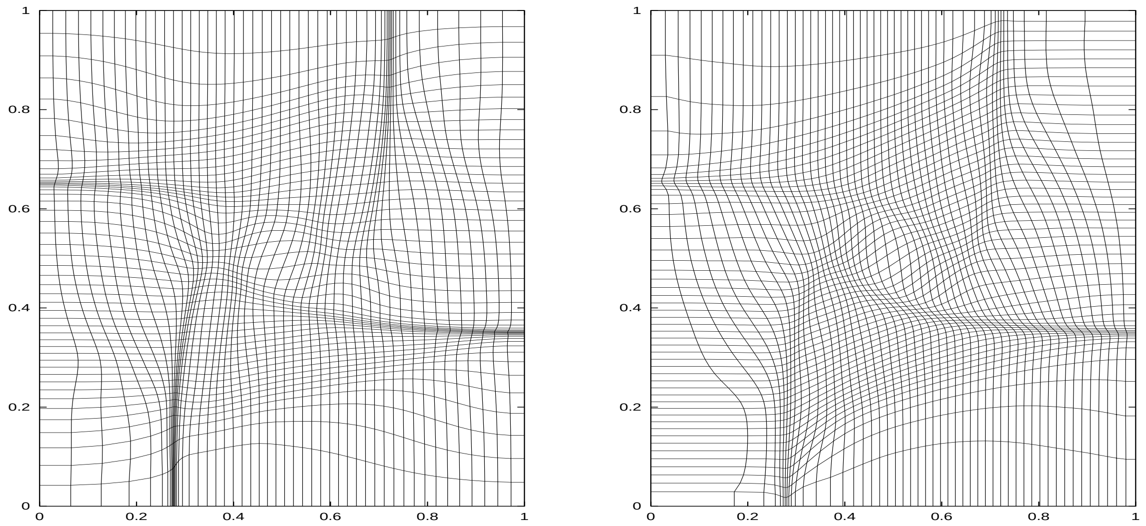

Figure 5: Example 3.2. Adaptive meshes at $t = 0.3$ with 50×50 grid points. Left: $\widehat{G}$ with $\beta_l = 8$ and $\gamma_l = 1$; Right: G with $\beta = 8$ and $\gamma = 1$.

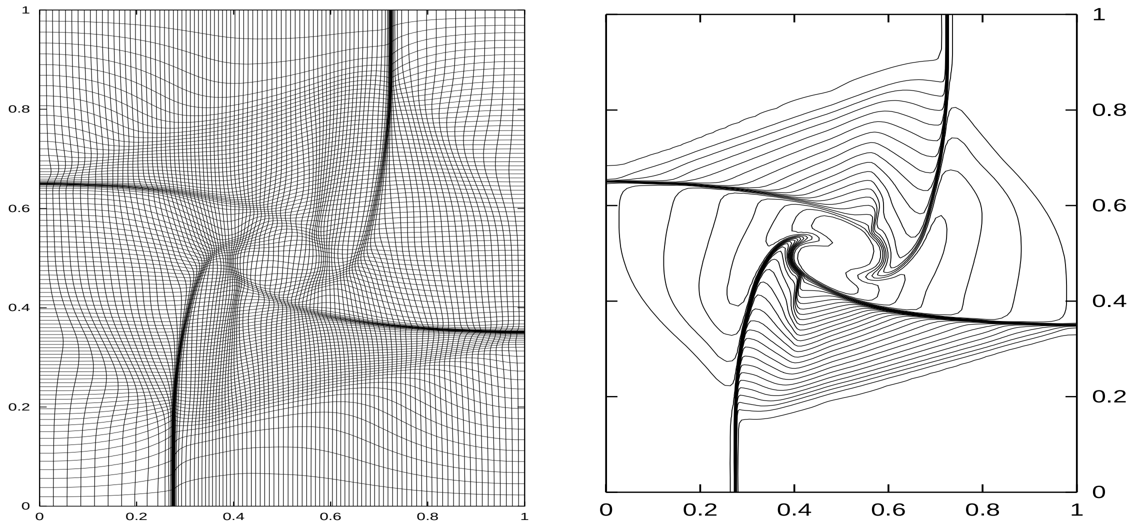

Figure 6: Example 3.2. The adaptive solutions at $t = 0.3$ with 100×100 grid points, obtained by using the monitor $\widehat{G}$ with $\beta_l = 8$ and $\gamma_l = 1$. Left: Adaptive mesh; Right: The density (30 equally contour lines).

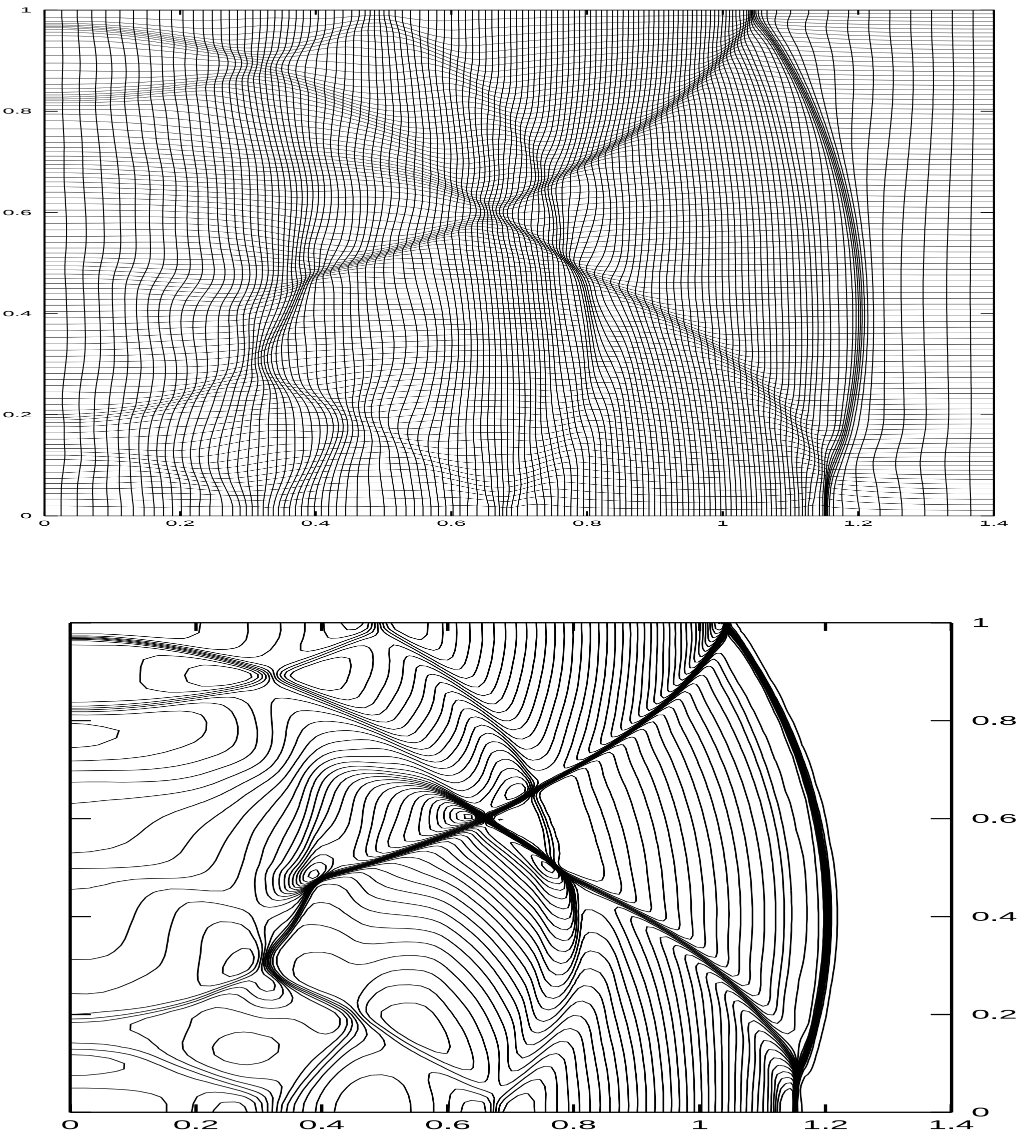

Figure 7: Example 3.3. The adaptive solutions at $t = 0.7$ with 125×100 grid points, obtained by using the monitor $\widehat{G}$ with $\beta_l = 10$ and $\gamma_l = 1$. Top: adaptive mesh; Bottom: the pressure (40 equally contour lines).

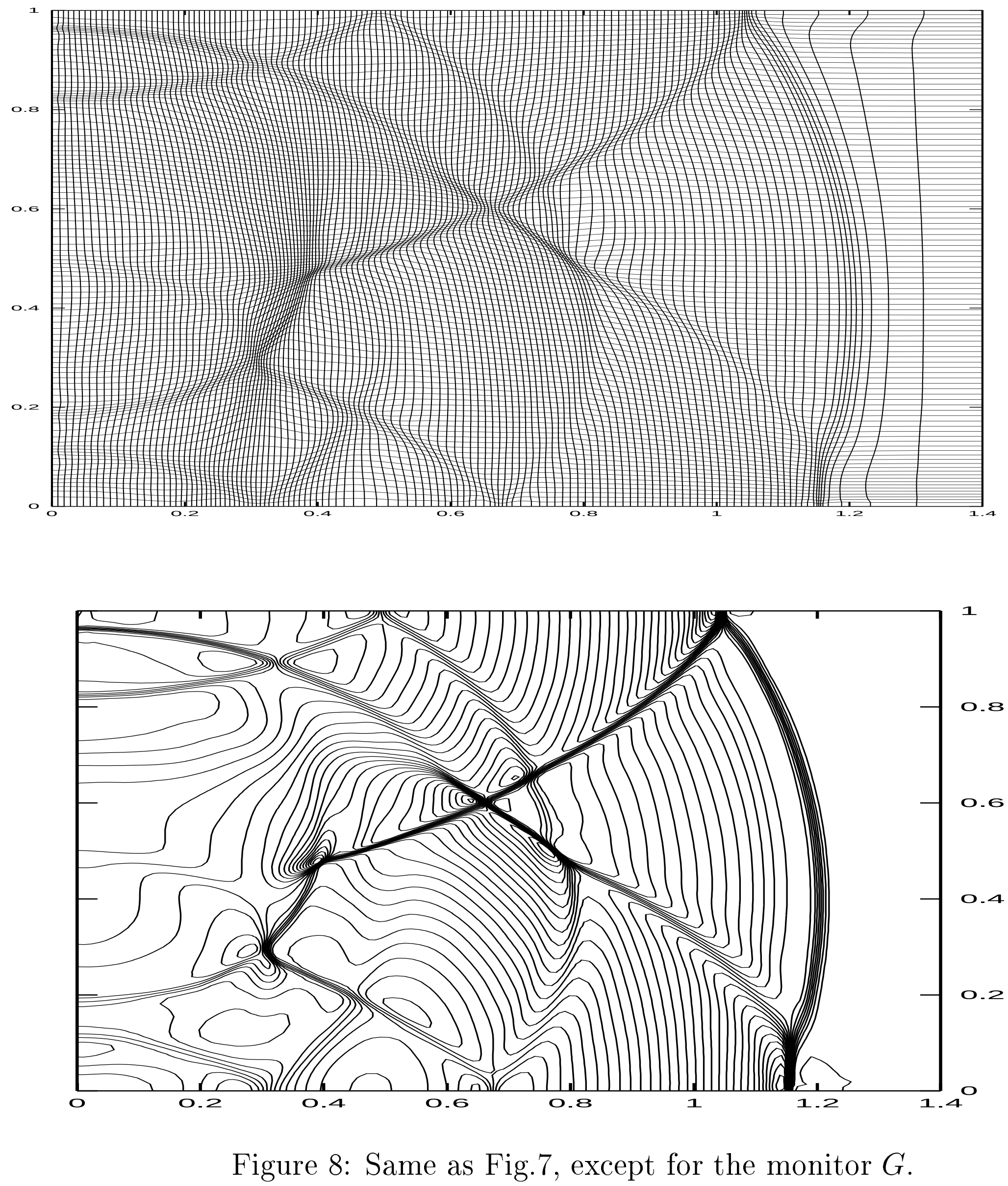

Figure 8: Same as Fig.7, except for the monitor G.

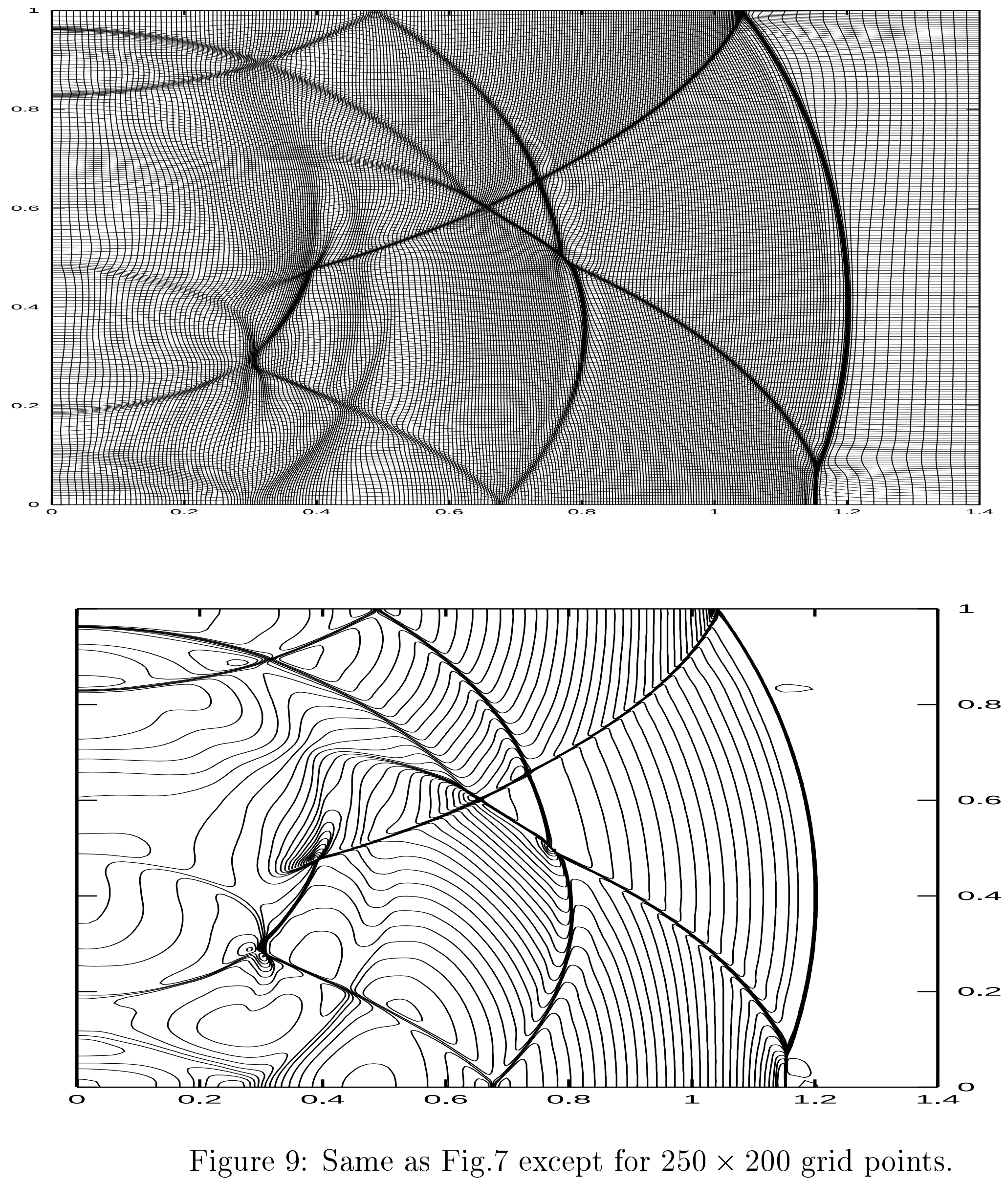

Figure 9: Same as Fig.7 except for 250×200 grid points.

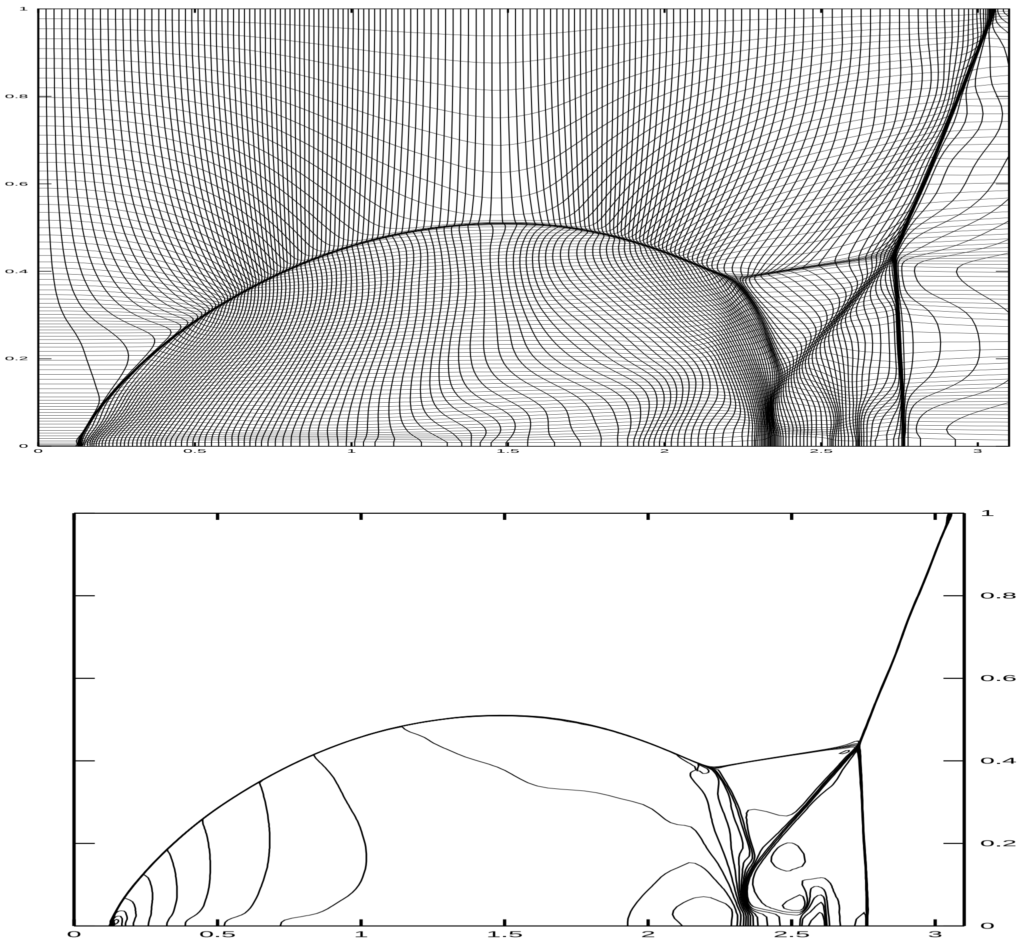

Figure 10: Example 3.4. The adaptive solutions at $t = 0.2$ with 160×80 grid points, obtained by using the monitor $\widehat{G}$ with $\beta_l = 0.6$ and $\gamma_l = 1$. Top: adaptive mesh; Bottom: the density (30 equally contour lines).

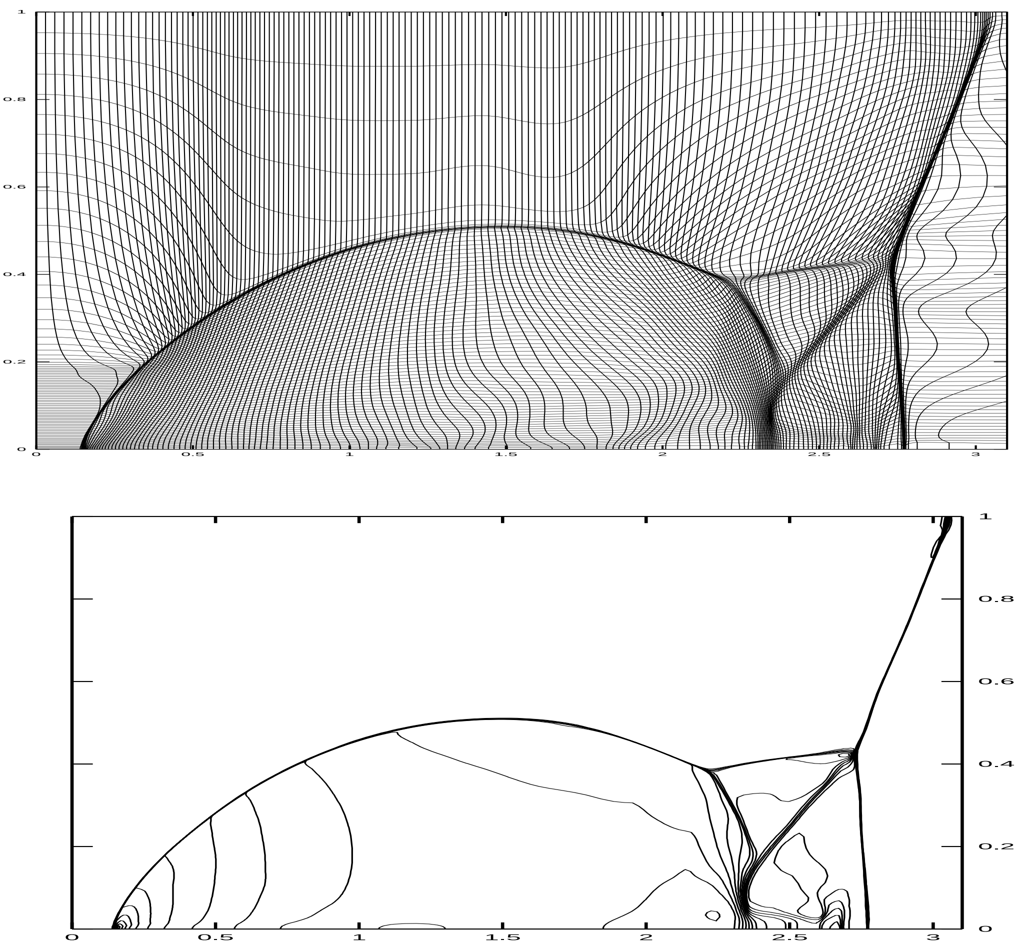

Figure 11: Same as Fig.10 except using the monitor G.

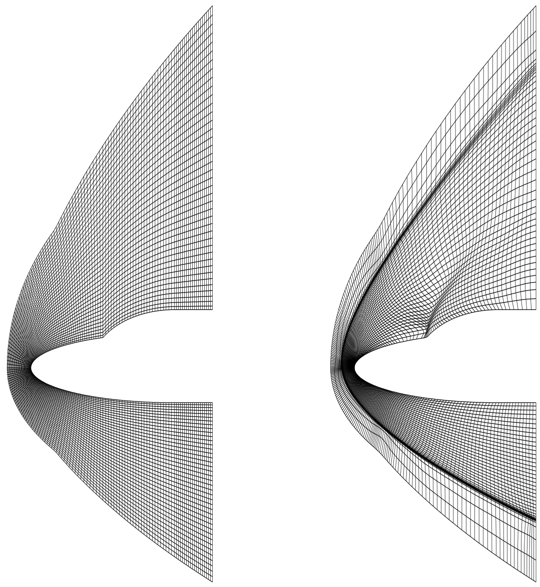

Figure 12: Example 3.5. The initial mesh (left) and the adaptive mesh (right) with 201×41 grid points.

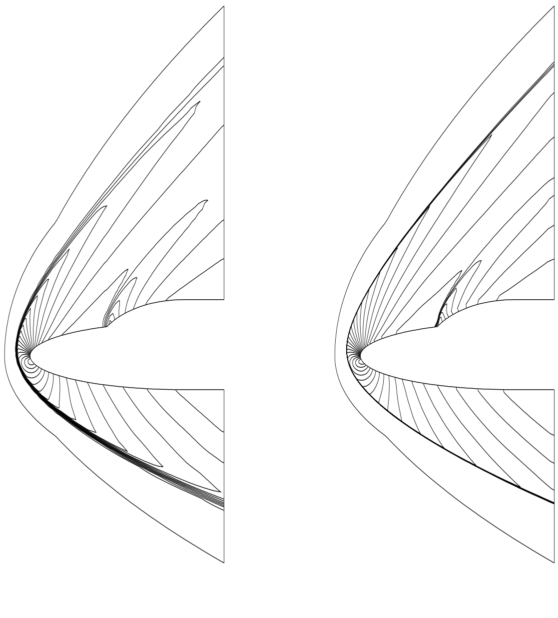

Figure 13: Example 3.5. The distribution of the presures with 22 equal contour lines obtained using the fixed mesh (left) and the adaptive mesh (right).