Robust and Efficient Adaptive Moving Mesh Solution of the 2-D Euler equations*

P. A. Zegeling & W. D. de Boer
Mathematical Institute, Utrecht University
P.O. Box 80.010, 3508 TA Utrecht, The Netherlands
H. Z. Tang
LMAM, School of Mathematical Sciences, Peking University
Beijing 100871, P.R. China

Abstract

In this paper we describe an adaptive moving mesh technique and its application to the 2D Euler equations. The adaptive mesh is derived from the minimization of a mesh-energy integral. A robust and efficient monitor function with a time-dependent, automatically chosen, adaptivity parameter is used to track individual features of the physical solutions, such as shocks and emerging instabilities. The results of a series of numerical experiments are presented, including shock waves and a Rayleigh-Taylor instability.

1 Introduction

Over the years a large number of adaptive mesh methods have been proposed for time-dependent partial differential equation (PDE) models. Here, we describe a dynamic-remeshing technique in which the nodes are moving continuously in the space-time domain. Examples of such methods can be found in [6, 11, 9, 15]. In this study an adaptive moving mesh method is derived that is based on a minimization of a so-called mesh-energy integral. The corresponding Euler-Lagrange equations then define a set of adaptive mesh PDEs. In one space dimension this reduces to the widely-used equi-distribution principle; in two space dimensions it is related (but not equivalent to) harmonic mapping based methods supplemented with a monitor matrix to detect the steep transitions in the solution.

*2000 Mathematical Subject Classification: 65Mxx, 65M50, 76L05, 76M20

Since the physical domain in our applications is rectangular, we can use a simplified system of adaptive mesh PDEs, similar to [11]. We have also borrowed ideas from [2] in which the difficult choice of a user-defined adaptivity constant in the monitor function is replaced by a procedure where the parameter is time-dependent and automatically chosen. Moreover, the discretization of the PDE model and the moving mesh is decoupled. For the numerical approximation of the physical PDE, which consists of systems of hyperbolic conservation laws, a finite volume discretization with a local Lax-Friedrichs numerical flux is chosen. For the resulting elliptic, adaptive mesh PDEs, a Gauß-Seidel type iteration method is used.

The layout of the paper is as follows. In the next section we present the 2D Euler equations model. The adaptive moving mesh method is defined in Section 3. Section 4 describes the numerical algorithm for solving the physical and mesh PDEs. In Section 5 numerical experiments are performed for several models with shocks and also a Rayleigh-Taylor instability model. Finally, Section 6 lists our conclusions.

2 The 2D Euler equations

The two-dimensional Euler equations of gas dynamics describing the behavior of an ideal compressible gas, are written in conservative form:

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ E\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ u(E+p)\end{pmatrix}+\frac{\partial}{\partial y}\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ v(E+p)\end{pmatrix}=\begin{pmatrix}0\\0\\0\\0\end{pmatrix} \tag{1}$$

where ρ is the density, $(\rho u, \rho v)^T$ is the momentum vector, E the total energy and p the pressure. Since we are working with an ideal gas, the equation of state, which shows how the energy is related to the pressure is provided as follows: $p = (\gamma - 1)(E - \rho\frac{u^2+v^2}{2})$, with γ the ratio of specific heats. Denote $\mathbf{r} := (x, y)$. In our computations, we will specify homogeneous Neumann conditions, except for one case that will be explained in Section 5.3. The initial data for each test case are given in Section 5.

3 The adaptive moving mesh PDEs

3.1 Winslow's method

The adaptive mesh can be seen as an approximation of a coordinate transformation between computational coordinates $(\xi, \eta)^T \in \Omega_c := [0, 1]\times[0, 1]$ (with a uniform mesh partitioning) and physical coordinates $(x,y)^T \in \Omega_p \subset \mathcal{R}^2$ (with a non-uniform, i.e. adaptive mesh). In a variational setting, a "mesh-energy" functional [14] can be defined as

$$\mathcal{E} = \frac{1}{2} \iint_{\Omega_c} \left(\nabla^T x \; \omega \; \nabla x + \nabla^T y \; \omega \; \nabla y \right) \quad d\xi d\eta, \tag{2}$$

where $\nabla = (\frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta})^T$ and $\omega > 0$ a monitor function in which the solution dependence of the transformation is reflected. Minimizing the energy $\mathcal{E}$, the corresponding Euler-Lagrange equations yield:

$$\nabla \cdot (\omega \nabla x) = 0, \qquad \nabla \cdot (\omega \nabla y) = 0. \tag{3}$$

These elliptic equations form the basis of the adaptive mesh algorithm. For the interested reader, we refer to [11, 15] for more details on system (3) and to [5, 6, 10] for more details on other types of monitors ω and functionals $\mathcal{E}$, respectively. In one space dimension this idea reduces to the well-known *equi-distribution principle*: $(\omega x_\xi)_\xi = 0$, with boundary conditions $x(0) = x_l, \quad x(1) = x_r$, or equivalently, $\xi_x = \mathrm{c} \; \omega$ with a constant c. Then, an explicit formula for ξ as a function of x can be easily derived:

$$1 = \xi(x_r) - \xi(x_l) = \int_{x_l}^{x_r} \xi_x \; d\bar{x} = c \int_{x_l}^{x_r} \omega \; d\bar{x} \Rightarrow \xi_x = \frac{\omega(x)}{\int_{x_l}^{x_r} \omega \; d\bar{x}},$$

and integrating once more gives $\xi(x) = \frac{\int_{x_l}^{x} \omega(\bar{x}) \; d\bar{x}}{\int_{x_l}^{x_r} \omega(\bar{x}) \; d\bar{x}}$.

It can be shown that the transformation, being the solution of (3), remains non-singular, which indicates that the adaptive mesh stays regular. The theoretical result that states the regularity was given by Clément et al [4]. Their proof consists of three main ingredients: application of the Carleman-Hartman-Winter theorem, the Jordan curve theorem and the maximum principle for elliptic PDEs. It reads:
Let $\omega \geq \tilde{c} > 0$, $\omega \in C^{0,1}(\Omega_c)$ and $\omega_\xi, \omega_\eta \in C^\gamma(\bar{\Omega}_c)$, $\gamma \in (0,1)$. With the boundary conditions $x|_{\xi=0} = y|_{\eta=0} = 0, x|_{\xi=1} = y|_{\eta=1} = 1, \frac{\partial x}{\partial n}|_{\xi=0} = \frac{\partial x}{\partial n}|_{\xi=1} = \frac{\partial y}{\partial n}|_{\eta=0} = \frac{\partial y}{\partial n}|_{\eta=1} = 0$, there exists a unique solution $(x,y) \in C^2(\bar{\Omega}_c)$,which is a bijection from $\bar{\Omega}_c$ into itself. Moreover, the Jacobian $\mathcal{J} = x_\xi y_\eta - x_\eta y_\xi > 0$, i.e. the transformation satisfying (3) is non-singular.

3.2 The monitor function ω

A traditional choice for the monitor function ω in (3) (see [11]) to detect regions with high spatial activity (for example, of the density ρ in the Euler equations) is an arclength-type monitor (AL-monitor)

$$\omega = \sqrt{1 + \alpha \ \nabla\rho \cdot \nabla\rho}. \tag{4}$$

Here the parameter α is an 'adaptivity'-parameter which controls the amount of adaptivity. For $\alpha = 0$, we have $\omega = 1$, representing a uniform mesh (the transformation becomes the identity). Higher values of $\alpha > 0$ allow for more adaptivity. However, α is problem-dependent: in general, there is no straightforward rule how to choose this parameter. It is known from practice that several experiments have to be done to arrive at an 'optimal' choice with respect to accuracy and efficiency.

An alternative for this monitor function deals with this issue and involves a *time-dependent* parameter that is *automatically chosen*. In the application of this paper it reads

$$\omega = \alpha(t) + ||\nabla\rho||_2^{\frac{1}{m}}, \quad \text{with} \ \ \alpha(t) = \iint_{\Omega_c} ||\nabla\rho||_2^{\frac{1}{m}} \ d\xi d\eta. \tag{5}$$

The choice $m = 2$ was made in [1] and [2] (for a slightly different case, namely using physical derivatives instead of computational derivatives in the monitor), following a 1D-result for a special situation. In this paper we consider $m = 1$. For this value of m, the derivatives of monitor (5) scale the same as for monitor (4), whereas for $m = 2$, they do not. Intuitively, the results of both monitor functions will be similar for $m = 1$. It is interesting to note that the idea of relating the adaptivity constant to the integral of the monitor values was already posed in [3]. However, the theoretical and practical implementation of this idea was only quite recently done in the earlier mentioned works.

Following ideas from [7, 5], we can give a short explanation why using (5) may be benificial. Define

$$\tilde{\omega} = 1 + \frac{\gamma ||\nabla\rho||_2}{(1-\gamma)\iint_{\Omega_c} ||\nabla\rho||_2 \ d\xi d\eta},$$

with $\gamma \in [0, 1)$. Then for $\gamma = \frac{1}{2}$ we have $\tilde{\omega} = 1 + \frac{||\nabla\rho||_2}{\iint_{\Omega_c} ||\nabla\rho||_2 \ d\xi d\eta}$ which is equivalent to using (5) in combination with (3), since the constant $\alpha(t)$ (at each time level) can be taken through the differentiation in (3). If, for example, $||\nabla\rho||_2 \approx \iint_{\Omega_c} ||\nabla\rho||_2 \ d\xi d\eta$, then $\tilde{\omega} \approx \frac{1}{1-\gamma}$, giving a uniform mesh. Furthermore, it can be derived that

$$\gamma = \frac{\iint_{\Omega_c} \tilde{\omega} \ d\xi d\eta \ - 1}{\iint_{\Omega_c} \tilde{\omega} \ d\xi d\eta}.$$

For increasing values of $\gamma \uparrow 1$ the mesh becomes more and more adaptive, whereas for $\gamma = \frac{1}{2}$ (our situation) we find that approximately half of the mesh points will be concentrated in regions of high spatial derivatives and the other half in more flat regions of the solution. This follows from the fact that $\iint_{\Omega_c} \tilde{\omega} \; d\xi d\eta$ is related to the total number of mesh points and $\iint_{\Omega_c} (\tilde{\omega} - 1) \; d\xi d\eta = \iint_{\Omega_c} \tilde{\omega} \; d\xi d\eta - 1$ to the number of mesh points in the steep layer (where $\tilde{\omega}$ is large), respectively. Therefore, an extra consequence of this property is that the adaptive meshes are much smoother distributed than for the constant α case in (4). With the choice (5), the use of applying an additional filter or smoother to the mesh or monitor values is not necessary. In literature, for instance in [11], smoother transitions in a non-uniform mesh can be obtained by working with the smoothed discretized value

$$\begin{aligned} \mathcal{S}(\omega_{i+\frac{1}{2},j+\frac{1}{2}}) =& \frac{1}{4}\omega_{i+\frac{1}{2},j+\frac{1}{2}} + \frac{1}{8}(\omega_{i+\frac{3}{2},j+\frac{1}{2}} + \omega_{i-\frac{1}{2},j+\frac{1}{2}} + \omega_{i+\frac{1}{2},j+\frac{3}{2}} + \omega_{i+\frac{1}{2},j-\frac{1}{2}}) \\ &+ \frac{1}{16}(\omega_{i-\frac{1}{2},j-\frac{1}{2}} + \omega_{i-\frac{1}{2},j+\frac{3}{2}} + \omega_{i+\frac{3}{2},j-\frac{1}{2}} + \omega_{i+\frac{3}{2},j+\frac{3}{2}}) \end{aligned} \tag{6}$$

instead of $\omega_{i,j}$ itself. This weighted sum corresponds to averaging the influence of the monitor values at neighbouring mesh points. In the numerical experiments we will denote this with *filter on* (using (6)), and *filter off* (working merely with ω_{ij} values).

4 The numerical PDE procedure

A possible approach, which is efficient in 1D, would be to couple the discretized systems for the adaptive mesh PDEs and the physical PDE. However, there are a number of disadvantages to this approach. One reason not to couple the PDEs is the following: a user may wish to control the discretization of the physical problem and such flexibility is severely restricted by coupling the unknowns together. We have chosen to decouple the numerical solution procedure for the Euler equations and adaptive mesh PDEs, solving for the mesh and the physical solution alternately. This will be briefly worked out in Section 4.4. First, we describe how to discretize the adaptive mesh PDEs.

4.1 Discretization of the mesh PDEs

Given a non-uniform partitioning $\{A_{i+\frac{1}{2},j+\frac{1}{2}}\}_{i,j}$ of the physical domain Ω_p, where $A_{i+\frac{1}{2},j+\frac{1}{2}}$ is a quadrangle with four vertices $\mathbf{r}_{i+k,j+l}$, $0 \leq k, l \leq 1$, as shown in Figure 1. Subdivide the computational domain $\Omega_c = \{(\xi,\eta)|\ 0 \leq \xi \leq 1,\ 0 \leq \eta \leq 1\}$ into the uniform mesh:

$$(\xi_i,\eta_j)|\ \xi_i = i\Delta\xi,\ \eta_j = j\Delta\eta;\ 0 \leq i \leq I_\xi + 1,\ 0 \leq j \leq I_\eta + 1$$

where $\Delta\xi = 1/(I_\xi + 1)$, $\Delta\eta = 1/(I_\eta + 1)$, and I_ξ and I_η are the number of mesh points in the $x-$ and $y-$direction. We characterize the numerical approximations to $\mathbf{r} = \mathbf{r}(\xi,\eta)$ by $\mathbf{r}_{i,j} = \mathbf{r}(\xi_i,\eta_j)$. The elliptic system of mesh PDEs is then discretized by second-order central finite differences in a straightforward manner. A Gauß-Seidel iteration method is used for the numerical solution of the resulting system of algebraic equations. The mesh points on the boundary of the rectangular domain are redistributed each time step by letting them move with the same speed as the tangential component of the mesh speed of the internal points adjacent to those boundary points (see [11] for more details on this).

4.2 A conservative solution-updating method

Having computed the new mesh as described in the previous section, the solution values have to be updated on this mesh by an interpolation method. In [11, 10] a conservative interpolation method is derived to preserve conservation of mass at each mesh redistribution step. Obviously, with simple linear interpolation this can not be achieved. Let $\mathbf{r}_{i,j}$ and $\tilde{\mathbf{r}}_{i,j}$ be the coordinates of old and new mesh points, respectively (the mesh point $\mathbf{r}_{i,j}$ moves to position $\tilde{\mathbf{r}}_{i,j}$ after having applied the Gauß-Seidel iterations). Similarly, denote with $A_{i+\frac{1}{2},j+\frac{1}{2}}$ and $\tilde{A}_{i+\frac{1}{2},j+\frac{1}{2}}$ the old and new finite control volumes, being set up as shown in figure 1. Then, using a perturbation technique (see [11]) and assuming small mesh speeds, it can be derived that the solution-updating scheme satisfies the following mass-conservation

$$\sum_{i,j} |\tilde{A}_{i+\frac{1}{2},j+\frac{1}{2}}|\ \tilde{\mathcal{Q}}_{i+\frac{1}{2},j+\frac{1}{2}} = \sum_{i,j} |A_{i+\frac{1}{2},j+\frac{1}{2}}|\ \mathcal{Q}_{i+\frac{1}{2},j+\frac{1}{2}},$$

where $|A|$ is the area of cell A, and $\mathcal{Q}$ and $\tilde{\mathcal{Q}}$ represent old and new numerical solution values in the physical PDE system.

4.3 Finite volume Discretization on non-uniform meshes

Consider the two-dimensional hyperbolic conservation laws

$$\frac{\partial \mathcal{Q}}{\partial t} + \frac{\partial \mathcal{F}_1(\mathcal{Q})}{\partial x} + \frac{\partial \mathcal{F}_2(\mathcal{Q})}{\partial y} = \mathcal{G}(x,y,\mathcal{Q}), \quad 0 < t \leq T, \tag{7}$$

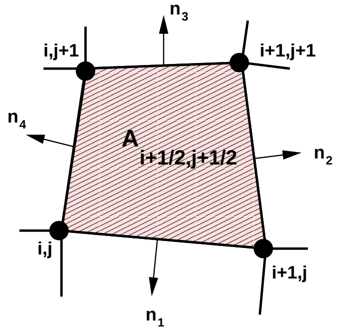

Figure 1: A typical non-uniform finite volume cell $A_{i+\frac{1}{2},j+\frac{1}{2}}$.

subject to the initial data $\mathcal{Q}(x, y, 0) = \mathcal{Q}_0(x, y)$, where T is the final time, and $\mathcal{Q}$ denotes the vector of conservative variables. $\mathcal{F}_1$ and $\mathcal{F}_2$ are the flux vectors in x- and y-direction, whereas $\mathcal{G}$ is a vector of source terms. A typical example of conservation laws in the form (7) with $\mathcal{G} = 0$ is given by the Euler equations as defined in (1). The case $\mathcal{G} \neq 0$ appears in the numerical experiments, when treating the Rayleigh-Taylor instability model in Section 5.3.

Assume that a mesh partitioning $\mathbf{r}_{i,j}$ of the physical domain Ω_p has been calculated. Then integrating (7) over the finite control volume $A_{i+\frac{1}{2},j+\frac{1}{2}}$ gives (see [12] or [13])

$$\frac{\partial}{\partial t}\iint_{A_{i+\frac{1}{2},j+\frac{1}{2}}} \mathcal{Q}\; dx\; dy + \sum_{l=1}^{4}\int_{s_l} \mathbf{F}_{\mathbf{n}^l}(\mathcal{Q})|_{(x,y)\in s_l}\; ds = \iint_{A_{i+\frac{1}{2},j+\frac{1}{2}}} \mathcal{G}\; dx\; dy,$$

where s_l $(l = 1, ...4)$ are the four boundary segments of the cell, $\mathbf{F}_{\mathbf{n}^l}(Q) = \mathcal{F}_1 n_x^l + \mathcal{F}_2 n_y^l$ and $\mathbf{n}^l = (n_x^l, n_y^l)^T$ $(l = 1, ..., 4)$ the normal outward vectors in the finite volume (see Figure 1). Assuming $\mathbf{F}_{\mathbf{n}^l} = \mathbf{F}^+_{\mathbf{n}^l} + \mathbf{F}^-_{\mathbf{n}^l}$, a general 2d finite volume scheme approximating (7) is given by

$$\begin{aligned}\mathcal{Q}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} =& \mathcal{Q}^{n}_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{\Delta t}{|A_{i+\frac{1}{2},j+\frac{1}{2}}|}\Big\{\mathcal{F}^-_{\mathbf{n}^1}(\mathcal{Q}^n_{i+\frac{1}{2},j-\frac{1}{2}}) + \mathcal{F}^-_{\mathbf{n}^2}(\mathcal{Q}^n_{i+3/2,j+\frac{1}{2}}) \\ &+ \mathcal{F}^-_{\mathbf{n}^3}(\mathcal{Q}^n_{i+\frac{1}{2},j+3/2}) + \mathcal{F}^-_{\mathbf{n}^4}(\mathcal{Q}^n_{i-\frac{1}{2},j+\frac{1}{2}}) + \sum_{l=1}^{4}\mathcal{F}^+_{\mathbf{n}^l}(\mathcal{Q}^n_{i+\frac{1}{2},j+\frac{1}{2}})\Big\} \\ &+ \Delta t\; \mathcal{G}^n_{i+\frac{1}{2},j+\frac{1}{2}}. \end{aligned} \tag{8}$$

In our experiments, we have used a local Lax-Friedrichs numerical flux in (8), and the initial data reconstruction to improve accuracy of the scheme (see [10]). The time step size Δt is determined every time step by $\Delta t = \frac{\min(\Delta x, \Delta y)\ CFL}{\max|\lambda|}$, where λ are the eigenvalues of the Jacobi matrix $\frac{\partial \mathbf{F}}{\partial \mathcal{Q}}$ and the CFL-number CFL will be specified in Section 5.

4.4 Decoupling of the PDEs

The steps in the full solution procedure can be summarized as follows:

Step 1 Partition the computational domain Ω_c uniformly and give an initial partition of the physical domain Ω_p; compute initial mesh values by a cell average of the control volume $A_{i+\frac{1}{2},j+\frac{1}{2}}$ based on the initial data $\mathcal{Q}(x, y, 0)$. In a loop over the time steps, update mesh and solution and evaluate the PDE.

Step 2a Move mesh $\mathbf{r}_{i,j}$ to $\tilde{\mathbf{r}}_{i,j}$ by solving the discretized mesh PDEs (3) using one Gauß-Seidel iteration.

Step 2b Compute the solution $\mathcal{Q}_{i+\frac{1}{2},j+\frac{1}{2}}$ on the new physical mesh based on the conservative interpolation from Section 4.2.
Repeat *step 2a* and *step 2b* for a fixed number of iterations.

Step 3 Evaluate the Euler equations by the finite volume method on the mesh $\tilde{\mathbf{r}}_{i,j}$ to obtain the solutions $\mathcal{Q}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}}$ at time-level t^{n+1}.

Step 4 Repeat *steps 2a, 2b* and *3* until the final point of time T has been reached.

5 Numerical experiments

In this section, we present several numerical experiments to demonstrate the performance of our algorithm. Until stated otherwise, the CFL number is taken as 0.48, $\gamma = 1.4$, and the cell number is 100×100 for our all computations. The double integral in (5) to compute the time-dependent parameter $\alpha(t)$ is approximated by applying the trapezoid rule at each time step. All experiments were performed on a SUNblade150 workstation with a sparcv9 processor (650 MHz).

5.1 2D Riemann problem: Shock waves

The first test example, which is a two-dimensional Riemann problem of the Euler equation, i.e. configuration 4 in [8], has the following initial data:

$$(\rho, u, v, p)_{t=0} = \begin{cases} (1.1, 0.0, 0.0, 1.1) & \text{if } x > 0.5,\ y > 0.5, \\ (0.5065, 0.8939, 0.0, 0.35) & \text{if } x < 0.5,\ y > 0.5, \\ (1.1, 0.8939, 0.8939, 1.1) & \text{if } x < 0.5,\ y < 0.5, \\ (0.5065, 0.0, 0.8939, 0.35) & \text{if } x > 0.5,\ y < 0.5. \end{cases} \tag{9}$$

They correspond to a left forward shock, right backward shock, upper backward shock and finally a lower forward shock. The spatial domain is $[0, 1] \times [0, 1]$ and the end point of time is $t = 0.25$.

We first check the dependence of the AL-monitor (4) on the value of the parameter α. In Figure 2, the computational results *with* the use of filter (6) are shown for $\alpha = 0.01, 2$, and 10, respectively. We have displayed 25 contour lines in each contour plot. As the parameter α increase, the shock resolution becomes sharper and the mesh is highly concentrated near the discontinuities. However, when α is chosen bigger, the computational costs become much higher (see also Table 1). Hence, a kind of 'optimal' choice, in terms of accuracy and efficiency, is $\alpha = 2$ (as chosen in [11]).

Next, we investigate the monitor function (5). The corresponding results shown in Figure 3 are computed by using the monitor (5) with $m = 2$ (the original choice) and $m = 1$ (a choice to fit better with the 'dimensions'). For comparison, the density on a uniform (non-adaptive) finer mesh is depicted there as well. Taking $m = 1$ instead of $m = 2$ gives also a slightly more concentrated mesh around the steep transitions in the numerical solution.

In Figure 4 we show adaptive meshes, zoomed in around the point (0.92, 0.15) to compare the effects of the different monitors and the filter. From left to right they are respectively AL-monitor with $\alpha = 2$ (filter off), AL-monitor with $\alpha = 2$ (filter on), and the monitor (5) with $m = 1$ (no filter needed). We observe that the straight shock in the first two cases becomes slightly curved due to the highly concentrated mesh there; on the other hand, the shape of the shock is still retained in the third case. Compared with numerical experiments on a very fine uniform mesh, we note that, in Table 1 and also in the left frames of Figure 3, the results could be improved by choosing a (much) smaller CFL number, yielding more accurate results, but then with even much longer computing times.

For an overview of the above described cases, all choices and runtimes are listed in Table 1.

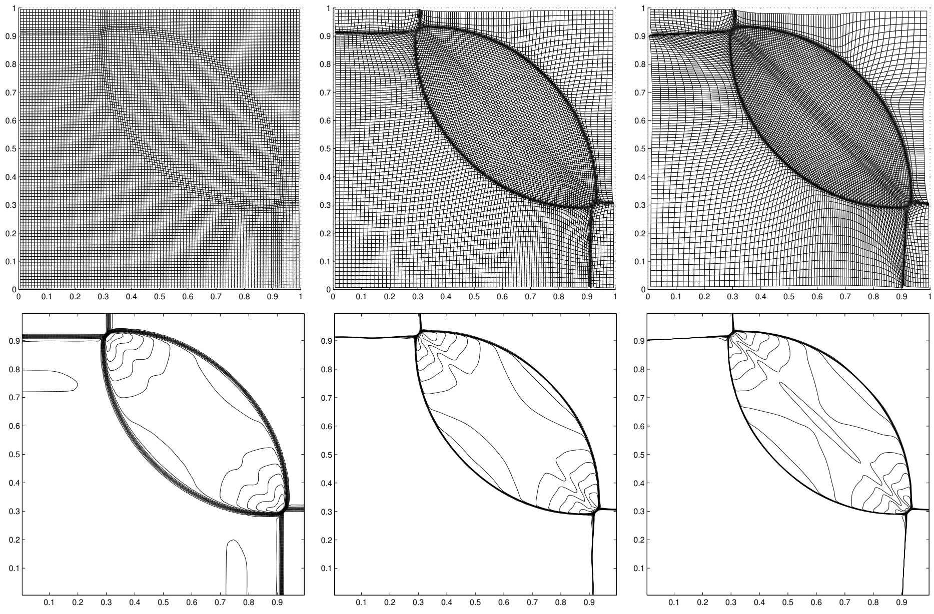

Figure 2: Numerical results for the AL-monitor for different values of α (left: $\alpha = 0.01$, middle: $\alpha = 2$, right: $\alpha = 10$).

5.2 Three other configurations

To further demonstrate the performance of the new adaptive mesh algorithm, we compute three other two-dimensional Riemann problems with the initial conditions corresponding to configurations 6, 8, and 12 named in [8]. The numerical solutions at $t = 0.25$, in terms of density contour lines, shown in Figure 5 are computed by using the monitor function (5) with $m = 1$ and no filter. The runtimes are : 0^h55^m for configuration 6, 0^h34^m for configuration 8 and 1^h25^m for configuration 12, respectively.

5.3 Rayleigh-Taylor instability

For the background of this model we refer to [16]. The instability happens on an interface between fluids with different when an acceleration is directed from the heavy fluid to the light fluid. It has a fingering nature, with bubbles of light fluid rising into the ambient heavy fluid and spikes of heavy fluid falling into the light fluid. For this case, we solve the two-dimensional Euler equations (1) with an extra source term: $\mathcal{G} = (0, 0, \rho g, \rho v g)^T$, where g is

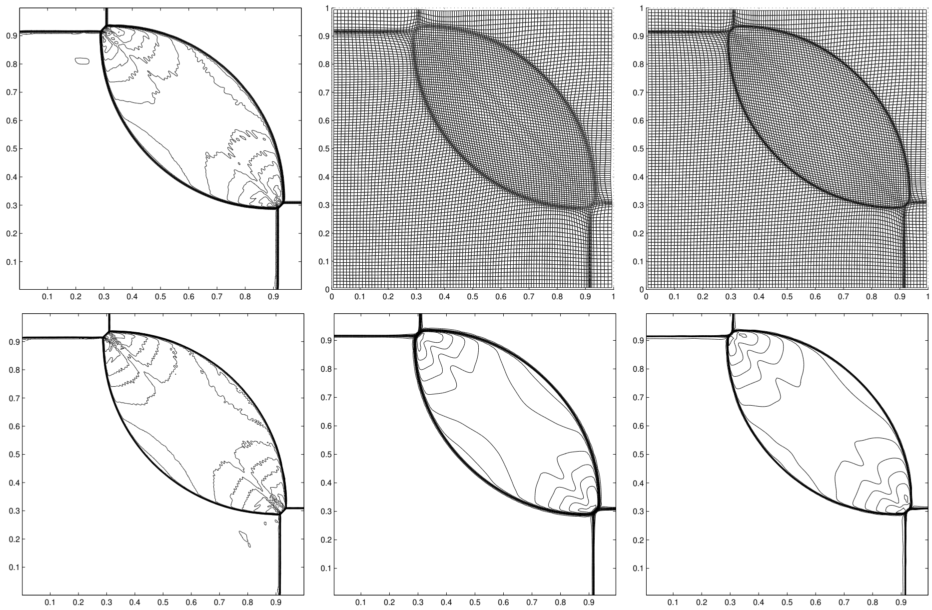

Figure 3: Numerical results for the monitor (5) (middle plot: $m = 2$, right plot: $m = 1$) and comparison with uniform mesh results (left upper plot: 400×400 mesh and left lower plot: 600×600 mesh).

the acceleration due to gravity. The computational domain is $[0, 0.25] \times [0, 1]$ and the interface is initially located at $y = 0.5$. Boundary conditions are implemented as follows: reflection conditions are at both left and right boundaries; the flow values are specified as $\rho = 1$, $p = 2.5$, u, $v = 0$ at the top, and $\rho = 2$, $p = 1$, $u, v = 0$ at the bottom. We now take $\gamma = \frac{5}{3}$ and $g = 1$ (a rescaled value). Figure 6 shows the computed solution (the density) and the adaptive mesh at four points of time. The results are obtained by using monitor (5) with $m = 1$ and no filter. We see that the mesh nicely concentrates near areas of high spatial activity. There is again no need for tuning of the numerical parameters in the adaptive method.

6 Conclusions

In this paper we have applied an adaptive moving mesh technique to the two-dimensional Euler equations from gas dynamics. The adaptive method, based on a coordinate transformation between physical and computational

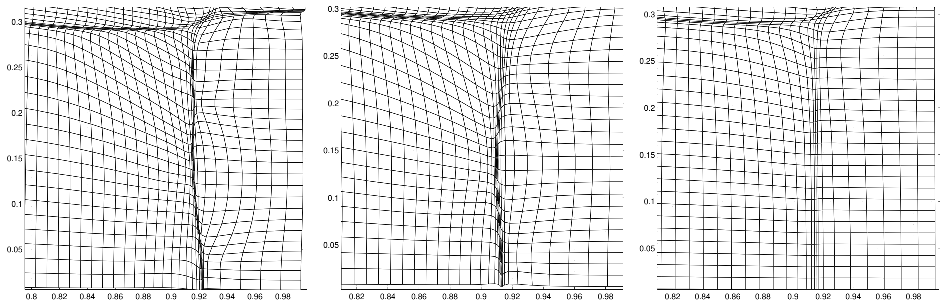

Figure 4: Adaptive mesh, zoomed in around the point (0.92, 0.15); left plot: AL-monitor with $\alpha = 2$ and filter off; middle plot: AL-monitor with $\alpha = 2$ and filter on; right plot: monitor (5) with $m = 1$ (no filter needed).

RUN #	*monitor*	α	m	*filter*	*runtime*
I	AL-monitor	0.01	-	on	0^h15^m
II	AL-monitor	0.1	-	on	0^h47^m
III	AL-monitor	2.0	-	on	2^h31^m
IV	AL-monitor	10.0	-	on	5^h48^m
V	AL-monitor	2.0	-	off	3^h04^m
VI	monitor (5)	-	2	off	0^h57^m
VII	monitor (5)	-	1	off	1^h09^m
VIII	uniform (400×400)	-	-	-	1^h05^m
IX	uniform (600×600)	-	-	-	3^h50^m

Table 1: The runtimes for the shock wave model (9).

coordinates, is derived from the minimization of a mesh-energy functional. Within the adaptive strategy a monitor function has been used with a *time-dependent* and *automatically chosen* parameter, in contrast to previous papers in which the adaptivity parameter had to be adjusted for each PDE model. Further, no additional filter was needed to deal with a potential non-smoothness of the nonuniform mesh. Moreover, the computing times for similar test models were much lower than when using the traditional arclength-type monitor function (*with* the need of tuning of the parameter *and* an extra filter for the mesh). The robustness and efficiency of the method has been demonstrated via several numerical experiments, among others, for shock waves and a Rayleigh-Taylor instability model.

Acknowledgments

The third author was partially supported by the Special Funds for Major State Basic Research Projects of China, Key Program of the National Natural Science Foundation of China, and the Alexander von Humboldt foundation.

References

[1] G. Beckett and J. A. Mackenzie, On a uniformly accurate finite difference approximation of a singularly perturbed reaction-diffusion problem using grid equidistribution, *Comput. Appl. Math.*, **131** (2001), 381-405.

[2] G. Beckett, J. A. Mackenzie, A. Ramage, and D. M. Sloan, Computational solution of two-dimensional unsteady PDEs using moving mesh methods, *J. Comput. Phys.*, **182** (2002), 478-495.

[3] J. G. Blom and P. A. Zegeling, Algorithm 731: A moving-grid interface for systems of one-dimensional time-dependent partial differential equations, *ACM Trans. in Math. Softw.*, **20** (1994), 194-214.

[4] P. Clément, R. Hagmeijer, and G. Sweers, On the invertibility of mappings arising in 2D grid generation problems, *Numer. Math.* **73** (1996), 37-51.

[5] W. Z. Huang and W. Sun, Variational mesh adaptation II: error estimates and monitor functions, *J. Comput. Phys.* **184** (2003), 619-648.

[6] W. Z. Huang and R. D. Russell, Moving mesh strategy based on a gradient flow equation for two-dimensional problems, *SIAM J. Sci. Comput.* **3** (1999), 998-1015.

[7] W. Z. Huang, L. Zheng, and X. Zhan, Adaptive moving mesh methods for simulating one-dimensional groundwater problems with sharp moving fronts, *Int. J. Num. Meth. in Eng.* **54** (2002), 1579-1603.

[8] 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), 319-340.

[9] 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), 811-825.

[10] H. Z. Tang, An moving mesh method for the Euler flow calculations using a directional monitor function, *preprint 2003*, available at: `http://www.math.ntnu.no/conservation/2003/031.html`.

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

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

[13] E. F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer, 1999.

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

[15] P. A. Zegeling, On resistive MHD models with adaptive moving meshes, to appear in *J. Sci. Comput.*, 2004.

[16] J. Shi, Y. Zhang, and C. W. Shu, Resolution of high order WENO schemes for complicated flow structures, *J. Comput. Phys.* **186** (2003), 690-696.

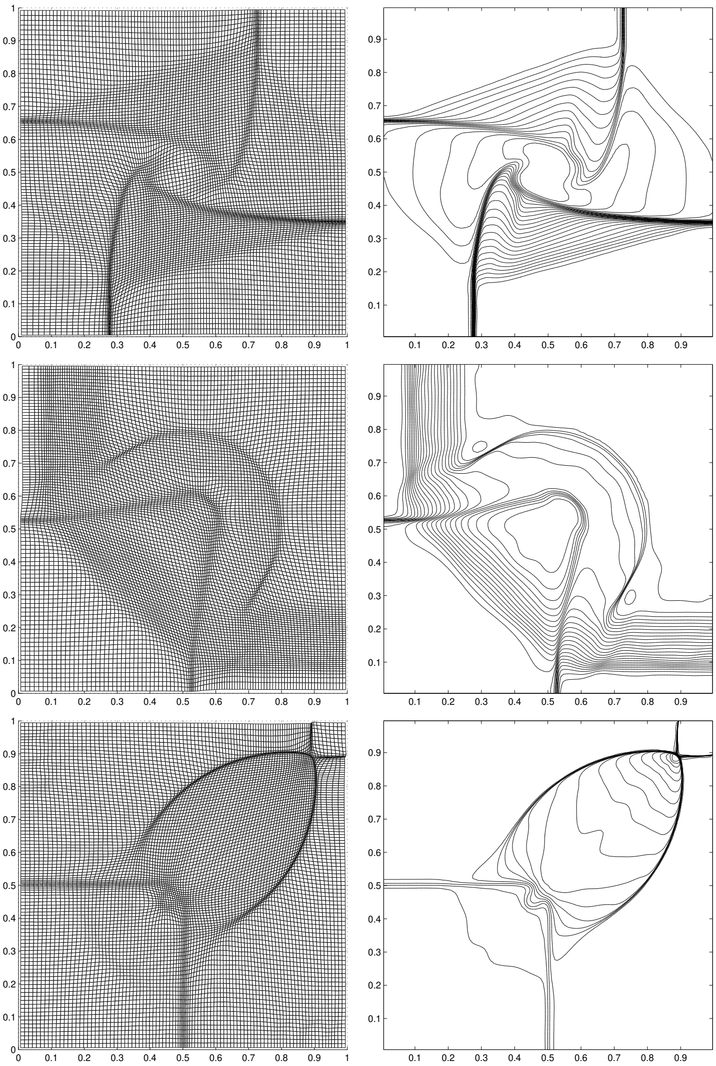

Figure 5: Numerical results for conf.6 (top), conf.8 (middle) and conf.12.

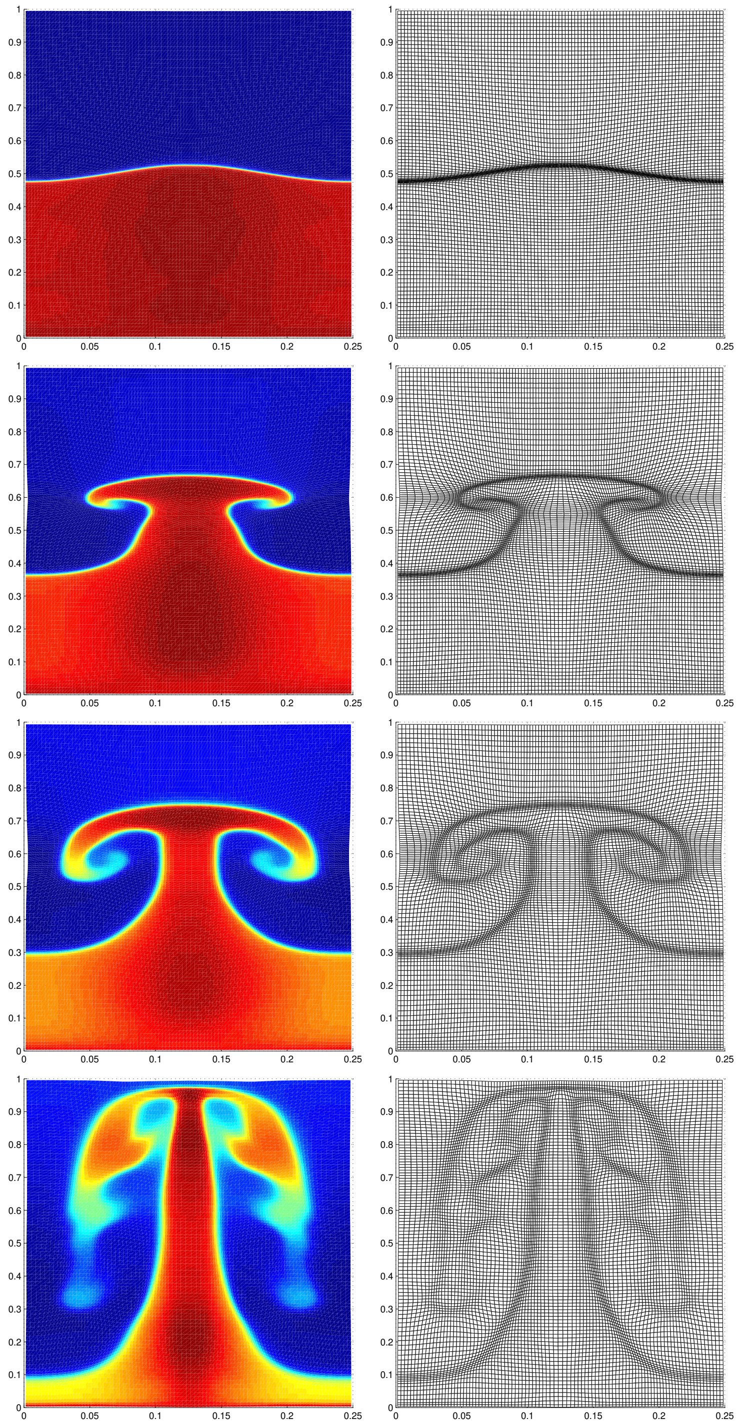

Figure 6: Numerical results for the Rayleigh-Taylor instability model at $t = 0.5, 1.5, 2.0$ and 3.0.