CONTEMPORARY
MATHEMATICS

330

Recent Advances in Scientific Computing and Partial Differential Equations

International Conference on the Occasion of
Stanley Osher's 60th Birthday
December 12–15, 2002
Hong Kong Baptist University, Hong Kong

S. Y. Cheng
C.-W. Shu
T. Tang
Editors

American Mathematical Society
Providence, Rhode Island

Contemporary Mathematics
Volume 330, 2003

Multi-Dimensional Moving Mesh Methods for Shock Computations

Huazhong Tang and Tao Tang

This paper is dedicated to Stan Osher on the occasion of his 60th birthday.

ABSTRACT. In this work, we demonstrate some recent progress on moving mesh methods with application to computational fluid dynamics. The emphasis will be on the application to the gas dynamics governed by hyperbolic system of conservation laws. Test problems in two- and three-dimensions are computed using a proposed moving mesh algorithm. The computations demonstrate our methods are efficient for solving problems with shock discontinuities.

1. Introduction

Adaptive mesh methods have important applications for a variety of physical and engineering areas such as solid and fluid dynamics, combustion, heat transfer, material science etc. It is especially true in areas where physical phenomena develop dynamically singular or nearly singular solutions in fairly localized regions, such as shock wave, detonation wave, blow-up and spike solutions, etc. The numerical investigation of these physical problems requires extremely fine meshes over a small portion of the physical domain to resolve accurately the large solution variations. In multi-dimensions, developing effective and robust adaptive grid methods for these problems becomes necessary. Successful implementation of the adaptive strategy can increase the accuracy of numerical solutions and also decrease the computational cost.

In the past two decades, there have seen many important progress in adaptive grid methods for solving partial differential equations (PDEs). Roughly speaking, there are two kinds of adaptive grid methods, static (or local refinement) and dynamic (or moving meshes). In the static adaptive methods, mesh points are moved at fixed time levels, where points can also be added to or removed from the mesh to obtain an approximate solution with the desired level of accuracy. The static

1991 *Mathematics Subject Classification.* 65M93, 35L64, 76N10.

Key words and phrases. Moving Mesh method, finite volume method, shock computation, hyperbolic conservation laws.

adaptive method is conceptually easy to apply and usually reliable. Many local refinement methods have been developed, see, e.g., [**4, 11, 12**] and references therein. However, the main disadvantage of the local refinement method is the complicated data structure and the choice of the error indicators. Recently, several more efficient error indicators were introduced for this purpose, see, e.g., [**9, 13, 14, 18**]. In particular, a detailed review of a posteriori error estimation is given in [**31**].

In the dynamic adaptive methods, the mesh points move continuously in time-space domain and concentrate in regions of large solution variations. For this type, a mesh equation is employed to move a mesh having a *fixed* number of nodes in such a way that the nodes remain concentrated in regions of rapid variation of the solution. It requires simple data structure and usually requires just a moderate number of spatial mesh-points. The moving mesh method is particularly useful when the underlying physical problems have very large (not moderate) singularities, e.g. shock, interior/boundary layers etc. In this case, the so-called monitor function (indicator) can be constructed quite easily. There are two classes of moving mesh methods. In the first class, the moving finite element method of Miller and Miller [**23**] and the moving finite difference method of Dorfi and Drury [**10**] have aroused considerable interest, see also [**7, 8, 21**]. In this approach, the mesh equation and the original differential equation are often solved simultaneously. In the second approach, the given PDEs and the mesh equations are decoupled, which are solved separately, see, e.g., [**1, 6, 19**].

The main objective of this paper is to develop a three-dimensional moving mesh method for shock capturing computations. Harten and Hyman [**15**] began the earliest study in this direction, by moving the grid at an adaptive speed in each time step to improve the resolution of shocks and contact discontinuities. After their work, many other moving mesh methods for hyperbolic problems have been proposed in the literature, including Azarenok et al. [**1, 2, 3**], Liu et al. [**22**], Mackenzie et al. [**26**] and Tang et al. [**27, 28**]. However, it is noticed that some 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. Multi-dimensional moving mesh methods are often difficult to use for fluid dynamics problems since the grid will typically suffer large distortions and possible tangling. It is therefore useful to design robust moving mesh algorithms for hyperbolic problems in multi-dimensions.

The paper is organized as follows. In Section 2, we outline the mesh generation procedure based on a variational approach. The numerical algorithm for moving mesh method is described in Section 3. Finite volume scheme for three-dimensional conservation laws will be outlined in Section 4. Numerical experiments are reported in Section 5. Finally, some concluding remarks are given in Section 6.

2. Mesh Generation Based on the Variational Approach

Let $\vec{r}_p = (x, y, z)$ and $\vec{r}_c = (\xi, \eta, \zeta)$ denote the physical and computational coordinates, respectively. A one-to-one coordinate transformation from the computational (or logical) domain Ω_c to the physical domain Ω_p is denoted by

$$\vec{r}_p = \vec{r}_p(\vec{r}_c), \quad \vec{r}_c \in \Omega_c. \tag{2.1}$$

Its inverse is denoted by

$$\vec{r}_c = \vec{r}_c(\vec{r}_p), \quad \vec{r}_p \in \Omega_p. \tag{2.2}$$

In the conventional variational approach, the mesh map is provided by the minimizer of a functional of the following form:

$$E[\vec{r}_c] = \frac{1}{2}\int_{\Omega_p} \left(\nabla\xi^T G^{-1}\nabla\xi + \nabla\eta^T G^{-1}\nabla\eta + \nabla\zeta^T G^{-1}\nabla\zeta\right)\, d\vec{r}_p, \tag{2.3}$$

where $\nabla := (\partial_x, \partial_y, \partial_z)$, G is a given symmetric positive definite matrix called *monitor function*. The variational mesh is determined by the Euler-Lagrange equations of the above functional

$$\nabla \cdot \left(G^{-1}\nabla\vec{r}_c\right) = 0. \tag{2.4}$$

If the monitor function G is taken as

$$G = \omega I, \tag{2.5}$$

then the mesh-generation method is the same as the well known Winslow's variable diffusion method [**32**]. In this paper, we will restrict our attention to the Winslow's approach. The extension to the general monitor function can be carried out in a straightforward way.

When the monitor (2.5) is used, the Euler-Lagrange equation (2.4) becomes

$$\partial_x\left(\frac{1}{w}\partial_x\vec{r}_c\right) + \partial_y\left(\frac{1}{w}\partial_y\vec{r}_c\right) + \partial_z\left(\frac{1}{w}\partial_z\vec{r}_c\right) = 0, \tag{2.6}$$

As observed by Tang et al. [**28**], directly solving the above elliptic system may be difficult, since the physical domain Ω_p may be of very complex geometry. On the other hand, it may be also difficult to solve the corresponding mesh generation equations obtained by interchanging the dependent and independent variables in (2.6) on the (simpler) computational domain, since the resulting equation is much more complicated than (2.6) (see, e.g., [**7, 19**]). To avoid these difficulties, we use a new energy functional

$$\widetilde{E}[\vec{r}_p] = \frac{1}{2}\int_{\Omega_c} \left(\widetilde{\nabla}^T xG\widetilde{\nabla}x + \widetilde{\nabla}^T yG\widetilde{\nabla}y + \widetilde{\nabla}^T zG\widetilde{\nabla}z\right) d\vec{r}_c, \tag{2.7}$$

to replace the standard functional (2.3), where G is again the monitor function, and $\widetilde{\nabla} = (\partial_\xi, \partial_\eta, \partial_\zeta)^T$. As a result, a new coordinate transformation $\vec{r}_p = \vec{r}_p(\vec{r}_c)$ can be obtained by solving the corresponding Euler-Lagrange equation

$$\widetilde{\nabla} \cdot \left(G\widetilde{\nabla}\vec{r}_p\right) = 0, \tag{2.8}$$

which results from minimizing the energy functional $\widetilde{E}[\vec{r}_p]$. When a scalar-type monitor (2.5) is used, the above equation becomes

$$\partial_\xi\left(\omega\partial_\xi\vec{r}_p\right) + \partial_\eta\left(\omega\partial_\eta\vec{r}_p\right) + \partial_\zeta\left(\omega\partial_\zeta\vec{r}_p\right) = 0. \tag{2.9}$$

Grid redistribution part in our 3D moving mesh method is to numerically solve the system (2.9). Since the indicator ω is in general associated with the underlying solutions to the given PDEs, the equation (2.9) is in fact a nonlinear elliptical equation which will be solved by an iteration strategy. Since the PDE solutions do not change too significantly at two consecutive time steps (due to small time stepping used), only a couple of iterations are sufficient to obtain a satisfactory solutions for (2.9). The detailed description on the iteration procedure can be found in [**19, 20, 27, 28**] for 1D and 2D moving mesh methods.

REMARK 2.1. More terms can be added to the functional (2.7) to control other properties of the mesh, such as orthogonality of the mesh and the alignment of the mesh lines with a prescribed vector field, as described by Brackbill et al. [5, 6].

REMARK 2.2. In 2D, Eq.(2.9) becomes

$$\partial_\xi \left(w \partial_\xi \vec{r}_p\right) + \partial_\eta \left(w \partial_\eta \vec{r}_p\right) = 0, \tag{2.10}$$

where $\vec{r}_p = (x, y)$. The boundary-point redistribution requires the use of a lower dimensional moving mesh method. The above equation can be used to serve for this purpose.

3. Algorithm for Moving Mesh Methods

Consider the three-dimensional hyperbolic system of conservation laws

$$\partial_t u + \partial_x F_x(u) + \partial_y F_y(u) + \partial_z F_z(u) = 0, \tag{3.1}$$

where F_x, F_y, F_z and u are vector functions in $\mathbf{R}^3$. The following steps describe the major steps to evolve the PDEs (3.1) and to redistribute the mesh grids at each time step.

Step 1: Initial data and grid preparation. Given an initial partition $(\vec{r}_p)^{[0]}_{i,j,k} := (\vec{r}_p)_{i,j,k}$ for the physical domain Ω_p and a uniform (fixed) partition for the logical domain Ω_c. Compute grid values $u^{[0]}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} := u((\vec{r}_p)_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}, t_0)$ based on the cell average of the initial data $u((\vec{r}_p), t_0)$ over the control volume $V_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}$ shown in Fig. 1, namely,

$$u\big((\vec{r}_p)_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}, t_0\big) = \frac{1}{|V_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}|} \iiint_{V_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}} u(x, y, z, t_0)\, dV. \tag{3.2}$$

Step 2: Mesh motion. Even at the initial stage, the grids should be redistributed based on the initial function, the initial partition and the corresponding grid values (3.2). Step 2 consists of three parts which are used not only for the grid re-distribution at the initial stage, but also for that at each later time-step. For $\nu = 0, 1, 2, \cdots$, do the following:

(a): Redistribute the grid points $\{(\vec{r}_p)^{[\nu]}_{i,j,k}\}$ to $\{(\vec{r}_p)^{[\nu+1]}_{i,j,k}\}$ by solving the mesh equation (2.9):

$$\begin{aligned} 0 = {} & \alpha_{i+\frac{1}{2},j,k}\left[(\vec{r}_p)^{[\nu]}_{i+1,j,k} - (\vec{r}_p)^{[\nu+1]}_{i,j,k}\right] - \alpha_{i-\frac{1}{2},j,k}\nabla_i(\vec{r}_p)^{[\nu+1]}_{i,j,k} \\ & +\beta_{i,j+\frac{1}{2},k}\left[(\vec{r}_p)^{[\nu]}_{i,j+1,k} - (\vec{r}_p)^{[\nu+1]}_{i,j,k}\right] - \beta_{i,j-\frac{1}{2},k}\nabla_j(\vec{r}_p)^{[\nu+1]}_{i,j,k} \\ & +\gamma_{i,j,k+\frac{1}{2}}\left[(\vec{r}_p)^{[\nu]}_{i,j,k+1} - (\vec{r}_p)^{[\nu+1]}_{i,j,k}\right] - \gamma_{i,j,k-\frac{1}{2}}\nabla_k(\vec{r}_p)^{[\nu+1]}_{i,j,k}, \end{aligned} \tag{3.3}$$

where ∇_l denote the backward difference operator in l-direction, and

$$\begin{aligned} \alpha_{i\pm\frac{1}{2},j,k} &= \tfrac{1}{4} \sum_{p,q=\pm\frac{1}{2}} \omega_{i\pm\frac{1}{2},j+p,k+q}, \\ \beta_{i,j\pm\frac{1}{2},k} &= \tfrac{1}{4} \sum_{p,q=\pm\frac{1}{2}} \omega_{i+p,j\pm\frac{1}{2},k+q}, \\ \gamma_{i,j,k\pm\frac{1}{2}} &= \tfrac{1}{4} \sum_{p,q=\pm\frac{1}{2}} \omega_{i+p,j+q,k\pm\frac{1}{2}}. \end{aligned}$$

(b): Update the solution values $\{u^{[\nu+1]}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\}$ over the new control volume $V^{[\nu+1]}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}$ using the following discrete conservative-interpolation:

$$
\begin{aligned}
|V^{[\nu+1]}_e| u^{[\nu+1]}_e &= |V^{[\nu]}_e| u^{[\nu]}_e - \sum_{l=1}^{6} (c_{\vec{n}_l} u)_{S^{[\nu]}_l} |S^{[\nu]}_l|, \\
e &= \{i+\tfrac{1}{2}, j+\tfrac{1}{2}, k+\tfrac{1}{2}\},
\end{aligned}
\tag{3.4}
$$

where $c_{\vec{n}_l} = \left((\vec{r}_p)^{[\nu]} - (\vec{r}_p)^{[\nu+1]}\right) \cdot \vec{n}_l$ denotes the component of the speed vector of mesh motion, $\vec{n}_l$ denotes outward unit normal vector of surface element S_l, $V^{[\nu+1]}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}$ and $V^{[\nu]}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}$ are two hexahedrons on new meshes $\{(\vec{r}_p)^{[\nu+1]}_{i,j,k}\}$ and old meshes $\{(\vec{r}_p)^{[\nu]}_{i,j,k}\}$, respectively, and $|\cdot|$ denote volume (or area) of volume (or surface) element. The use of (3.4) in 1D is justified by the authors in [**27**]. Note that the above solution-updating method guarantees the conservation of mass in the following sense:

$$
\sum_{i,j,k} |V^{[\nu+1]}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}| u^{[\nu+1]}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = \sum_{i,j,k} |V^{[\nu]}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}| u^{[\nu]}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}.
\tag{3.5}
$$

(c): Repeat the updating procedure (a) and (b) until $\|(\vec{r}_p)^{[\nu+1]} - (\vec{r}_p)^{[\nu]}\|$ is smaller than a given tolerance.

Step 3: Evolve the system of conservation laws (3.1) by using high resolution finite volume methods on the mesh $\{(\vec{r}_p)^{[\nu+1]}_{i,j,k}\}$ to obtain the numerical approximations $u^{n+1}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}$ at the time level t_{n+1}. The high resolution finite volume methods will be described in next section.

Step 4: If $t_{n+1} \le T$, then let $u^{[0]}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} := u^{n+1}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}$ and $(\vec{r}_p)^{[0]}_{i,j,k} := (\vec{r}_p)^{[\nu+1]}_{i,j,k}$ and go to Step 2.

Two issues require some attention. First, some temporal or spatial smoothing on the monitor function should be used to obtain smoother meshes. One of the reasons for using smoothing is to avoid very singular mesh and/or large approximation error around the stiff solution areas. In this work, we apply the following low pass filter to smooth the monitor

$$
\begin{aligned}
\omega_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \leftarrow\ & \frac{1}{8}\omega_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} + \frac{1}{16}\big(\omega_{i+\frac{3}{2},j+\frac{1}{2},k+\frac{1}{2}} + \omega_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \\
& +\omega_{i+\frac{1}{2},j+\frac{3}{2},k+\frac{1}{2}} + \omega_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}} + \omega_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{3}{2}} + \omega_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}\big) \\
& + \frac{1}{32}\big(\omega_{i+\frac{3}{2},j+\frac{3}{2},k+\frac{1}{2}} + \omega_{i+\frac{3}{2},j+\frac{1}{2},k+\frac{3}{2}} + \omega_{i+\frac{1}{2},j+\frac{3}{2},k+\frac{3}{2}} + \omega_{i-\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}} \\
& +\omega_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + \omega_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + \omega_{i+\frac{3}{2},j-\frac{1}{2},k+\frac{1}{2}} + \omega_{i+\frac{3}{2},j+\frac{1}{2},k-\frac{1}{2}} \\
& +\omega_{i-\frac{1}{2},j+\frac{3}{2},k+\frac{1}{2}} + \omega_{i+\frac{1}{2},j+\frac{3}{2},k-\frac{1}{2}} + \omega_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{3}{2}} + \omega_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{3}{2}}\big) \\
& + \frac{1}{64}\big(\omega_{i+\frac{3}{2},j+\frac{3}{2},k+\frac{3}{2}} + \omega_{i+\frac{3}{2},j+\frac{3}{2},k-\frac{1}{2}} + \omega_{i+\frac{3}{2},j-\frac{1}{2},k+\frac{3}{2}} + \omega_{i+\frac{3}{2},j-\frac{1}{2},k-\frac{1}{2}} \\
& +\omega_{i-\frac{1}{2},j+\frac{3}{2},k+\frac{3}{2}} + \omega_{i-\frac{1}{2},j+\frac{3}{2},k-\frac{1}{2}} + \omega_{i-\frac{1}{2},j-\frac{1}{2},k+\frac{3}{2}} + \omega_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}\big),
\end{aligned}
$$

where $\omega_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = \omega\big(u_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}, (\nabla u)_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\big)$. The second issue is about the redistribution of boundary grid points. Actually, the 3D boundary point redistribution can be regarded as local two-dimensional mesh motion. More precisely, the 2D mesh equation (2.9) can be solved to obtain the boundary point redistribution.

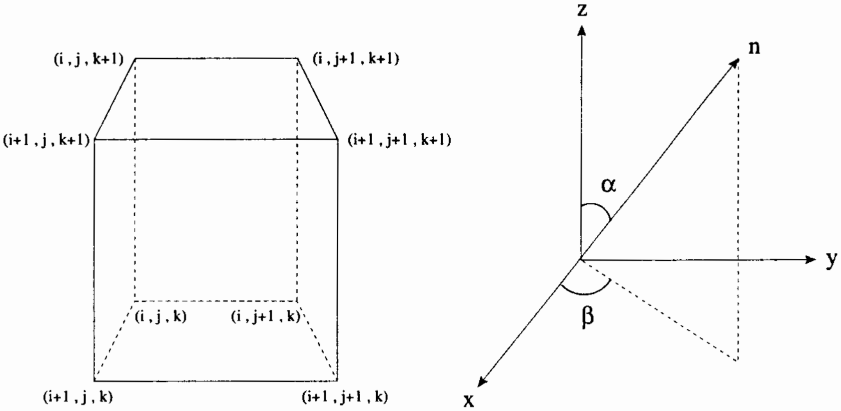

FIGURE 1. Three-dimensional control volume $V_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}$ and vector $\vec{n}$.

4. Finite Volume Method for Conservation Laws

Our solution procedure is based on two independent parts: a mesh-redistribution algorithm and a solution algorithm. As described in the last section the first part is based on an iteration procedure and the second part is independent of the first one. In principle, it can be any of the standard codes for solving the given PDEs, such as ENO schemes [**25**], central schemes [**24**] and BGK schemes [**33**]. In this paper, the underlying PDEs are solved by a high resolution finite volume method with a kinetic flux-vector splitting (KFVS) approach.

4.1. Scalar equation. Consider the three-dimensional scalar hyperbolic conservation laws

$$\partial_t u + \partial_x f_x(u) + \partial_y f_y(u) + \partial_z f_z(u) = 0. \tag{4.1}$$

Integrating the above equation over an arbitrary three-dimensional control volume V and using the divergence theorem give

$$\partial_t \iiint_V u \, dV + \iint_{\partial V} f_{\vec{n}}(u) \, dS = 0, \tag{4.2}$$

where $f_{\vec{n}}(u) = \vec{f} \cdot \vec{n}$, $\vec{f} = (f_x, f_y, f_z)$, and $\vec{n} = (n_x, n_y, n_y)$ denotes outward unit normal vector of surface element dS. Particularly, we assume that V is a finite hexahedron (See Fig. 1) with eight vertices: $\mathrm{A}_1((\vec{r}_p)_{i,j,k})$, $\mathrm{B}_1((\vec{r}_p)_{i+1,j,k})$, $\mathrm{C}_1((\vec{r}_p)_{i+1,j+1,k})$, $\mathrm{D}_1((\vec{r}_p)_{i,j+1,k})$, $\mathrm{A}_2((\vec{r}_p)_{i,j,k+1})$, $\mathrm{B}_2((\vec{r}_p)_{i+1,j,k+1}$, $\mathrm{C}_2((\vec{r}_p)_{i+1,j+1,k+1})$, and $\mathrm{D}_2((\vec{r}_p)_{i,j+1,k+1})$. Thus the bounding surface ∂V of the control volume V consists of six sub-face elements S_l, $1 \le l \le 6$:

$$\begin{array}{lll} S_1 := \mathrm{A}_1\mathrm{B}_1\mathrm{C}_1\mathrm{D}_1, & S_2 := \mathrm{A}_2\mathrm{B}_2\mathrm{C}_2\mathrm{D}_2, & S_3 := \mathrm{A}_1\mathrm{B}_1\mathrm{B}_2\mathrm{A}_2, \\ S_4 := \mathrm{C}_1\mathrm{D}_1\mathrm{D}_2\mathrm{C}_2, & S_5 := \mathrm{A}_1\mathrm{D}_1\mathrm{D}_2\mathrm{A}_2, & S_6 := \mathrm{C}_1\mathrm{B}_1\mathrm{B}_2\mathrm{C}_2. \end{array}$$

The explicit finite volume method for scalar conservation laws (4.1) is then of the form

$$u^{n+1}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = u^{n}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - \frac{\Delta t}{|V_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}|} \sum_{l=1}^{6} \hat{f}_{\vec{n}_l} |S_l| \tag{4.3}$$

where $\hat{f}_{\vec{n}_l} = \hat{f}(u^R_{\vec{n}_l}, u^L_{\vec{n}_l})$ is an appropriate numerical flux approximating $f_{\vec{n}_l}(u)$ at the cell interface S_l, and $u^L_{\vec{n}_l}$ and $u^R_{\vec{n}_l}$ denote the approximation of cell averages of u over the left and right control volume of the surface S_l along the corresponding outward normal direction, respectively. As an example, $\hat{f}_{\vec{n}_l}$ can take the following Lax-Friedrich flux:

$$\hat{f}_{\vec{n}_l} = \frac{1}{2}\left(f_{\vec{n}_l}(u^R_{\vec{n}_l}) + f_{\vec{n}_l}(u^L_{\vec{n}_l})\right) - \frac{1}{2}\max\{|\partial_u f_{\vec{n}_l}(u)|\}\left(u^R_{\vec{n}_l} - u^L_{\vec{n}_l}\right). \tag{4.4}$$

4.2. The Euler equations. The three-dimensional compressible Euler equations for inviscid gas dynamics may be written in the following form

$$\partial_t U + \partial_x F_x(U) + \partial_y F_y(U) + \partial_z F_z(U) = 0, \tag{4.5}$$

where

$$U = \begin{pmatrix} \rho \\ \rho u_x \\ \rho u_y \\ \rho u_z \\ E \end{pmatrix}, \quad F_x = \begin{pmatrix} \rho u_x \\ \rho u_x^2 + p \\ \rho u_x u_y \\ \rho u_x u_z \\ u_x(E+p) \end{pmatrix}, \quad F_y = \begin{pmatrix} \rho u_y \\ \rho u_y u_x \\ \rho u_y^2 + p \\ \rho u_y u_z \\ u_y(E+p) \end{pmatrix}, \quad F_z = \begin{pmatrix} \rho u_z \\ \rho u_z u_x \\ \rho u_z u_y \\ \rho u_z^2 + p \\ u_z(E+p) \end{pmatrix}.$$

Here ρ, $\vec{u} = (u_x, u_y, u_z)$, p, and $E = \rho e + \frac{1}{2}\rho\vec{u}^2$ are the density, velocity vector, pressure, and the total energy, respectively. To close the above system, we use the equation of state for ideal gases, $p = (\gamma - 1)\rho e$, where γ is the ratio of specified heat capacities of the fluid, and e denotes the internal energy density.

In this paper, we use a high resolution finite volume method with a kinetic flux-vector splitting (KFVS) approach to solve the three-dimensional Euler equations (4.5). The explicit 3D KFVS finite volume method can be written in the same form of (4.3):

$$U^{n+1}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = U^n_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - \frac{\Delta t}{|V_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}|}\sum_{l=1}^{6} \hat{F}_{\vec{n}_l}|S_l|,$$

where numerical flux $\hat{F}_{\vec{n}_l}$ is of the following form [29]:

$$\hat{F}_{\vec{n}_l} = F^+_{\vec{n}_l}(U^{L,n}_{\vec{n}_l}) + F^-_{\vec{n}_l}(U^{R,n}_{\vec{n}_l}), \qquad 1 \le l \le 6. \tag{4.6}$$

The split flux vectors $F^{\pm}_{\vec{n}_l}$ are given by

$$F^{\pm}_{\vec{n}_l}(U) = \rho M \begin{pmatrix} \langle v^1_{\vec{n}_l}\rangle_{\pm} \\ \langle v^2_{\vec{n}_l}\rangle_{\pm} \\ \langle v^1_{\vec{n}_l}\rangle_{\pm}\left\langle v^1_{\vec{t}_l}\right\rangle \\ \langle v^1_{\vec{n}_l}\rangle_{\pm}\left\langle v^1_{\vec{\tau}_l}\right\rangle \\ \frac{1}{2}\left[\langle v^3_{\vec{n}_l}\rangle_{\pm} + \langle v^1_{\vec{n}_l}\rangle_{\pm}\left(\left\langle v^2_{\vec{t}_l}\right\rangle + \langle v^2_{\vec{\tau}_l}\rangle + \langle \xi^2\rangle\right)\right] \end{pmatrix}, \tag{4.7}$$

where $\vec{t}_l$ and $\vec{\tau}_l$ are tangent vectors of surface S_l perpendicular to each other, $\langle\cdot\rangle$ and $\langle\cdot\rangle_{\pm}$ are some fundamental moments which can be found in [29], and M is a local rotating transformation

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \sin(\alpha)\cos(\beta) & -\sin(\beta) & -\cos(\alpha)\cos(\beta) & 0 \\ 0 & \sin(\alpha)\sin(\beta) & \cos(\beta) & -\cos(\alpha)\sin(\beta) & 0 \\ 0 & \cos(\alpha) & 0 & \sin(\alpha) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Here α and β denote the angle between the vector $\vec{n}_l$ and z- and x-axises, respectively, as illustrated in Fig. 1. For a first-order scheme, we can use

$$\begin{aligned} U_{\vec{n}_l}^{L,n} &= U_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}, & U_{\vec{n}_1}^{R,n} &= U_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}, \\ U_{\vec{n}_2}^{R,n} &= U_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{3}{2}}, & U_{\vec{n}_3}^{R,n} &= U_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}}, \\ U_{\vec{n}_4}^{R,n} &= U_{i+\frac{1}{2},j+\frac{3}{2},k+\frac{1}{2}}, & U_{\vec{n}_5}^{R,n} &= U_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}, \\ U_{\vec{n}_6}^{R,n} &= U_{i+\frac{3}{2},j+\frac{1}{2},k+\frac{1}{2}}, \end{aligned} \tag{4.8}$$

To obtain a second-order scheme, a piecewise linear function can be employed to replace the piecewise constant approach, in a way similar to the van Leer's initial reconstruction technique [**30**]. For example, for a regular grid system, we can use

$$\begin{aligned} U_{\vec{n}_1}^{L,n} &= U_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - \frac{\Delta z}{2}(U_z)_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}, \\ U_{\vec{n}_1}^{R,n} &= U_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + \frac{\Delta z}{2}(U_z)_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}}. \end{aligned} \tag{4.9}$$

5. Numerical Tests

5.1. 2D computations. For completeness, we include in this section also a 2D test. The procedures described in earlier sections can be easily adopted to the 2D case. More computational results in two space dimensions can be found in Tang and Tang [**27**].

EXAMPLE 5.1. **A 2D Riemann problem.** Two-dimensional Euler equations of gas dynamics can be written as

$$\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}_t + \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u(E+p) \end{bmatrix}_x + \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v(E+p) \end{bmatrix}_y = 0, \tag{5.1}$$

where ρ, (u, v), p, and E are density, velocity, pressure, and total energy, respectively. For an ideal gas, the equation of state is $p = (\gamma - 1)(E - \rho(u^2 + v^2)/2)$. The initial data is chosen as

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

which corresponds to the case of left forward shock, right backward shock, upper backward shock, and lower forward shock.

In [**17**], Lax and Liu computed 2D Riemann problems with various initial data using the positive schemes. The problem considered here corresponds to the Configuration 4 discussed in their paper. We use our adaptive mesh algorithm with 50×50 and 100×100 grid points to compute this Riemann problem and display the meshes and density at $t = 0.25$ in Fig. 2. It is found that our results with 100×100 mesh points give sharper shock resolution than that of the positive schemes with 400×400 grid points (see [**17**], p. 333). The monitor function used in this computation is $G = \omega I$ with $\omega = \sqrt{1 + 2(\rho_\xi^2 + \rho_\eta^2)}$.

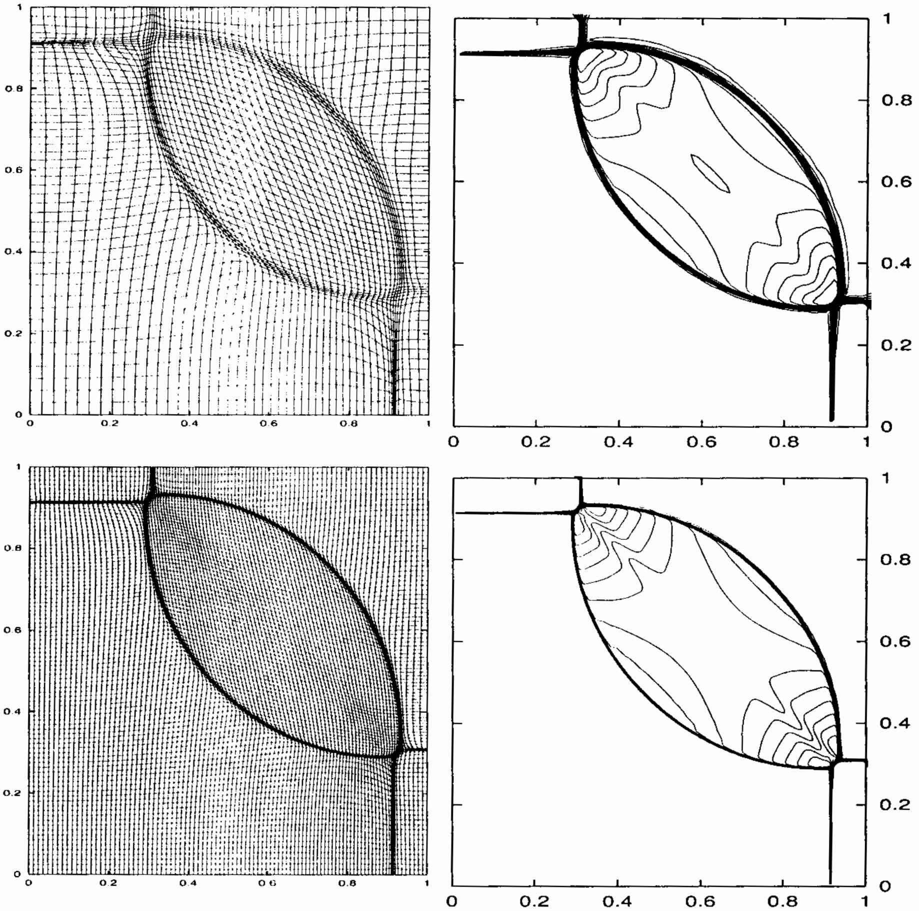

FIGURE 2. Example 5.1: The contours of mesh (left) and the density (right) with 50×50 grid points (top) and 100×100 grid points (bottom). 30 equally spaced contour lines are used for the density.

5.2. 3D computations.

EXAMPLE 5.2. **Linear advection problem.** The first 3D problem is the scalar advection equation

$$\partial_t \rho + \partial_x(\rho u_x) + \partial_y(\rho u_y) + \partial_z(\rho u_z) = 0, \tag{5.2}$$

where the velocity field is given by

$$\begin{aligned} u_x(x, y, z, t) &= 2\sin^2(\pi x)\sin(\pi y)\sin(\pi z)g(t), \\ u_y(x, y, z, t) &= -\sin(\pi x)\sin^2(\pi y)\sin(\pi z)g(t), \\ u_z(x, y, z, t) &= -\sin(\pi x)\sin(\pi y)\sin^2(\pi z)g(t), \end{aligned}$$

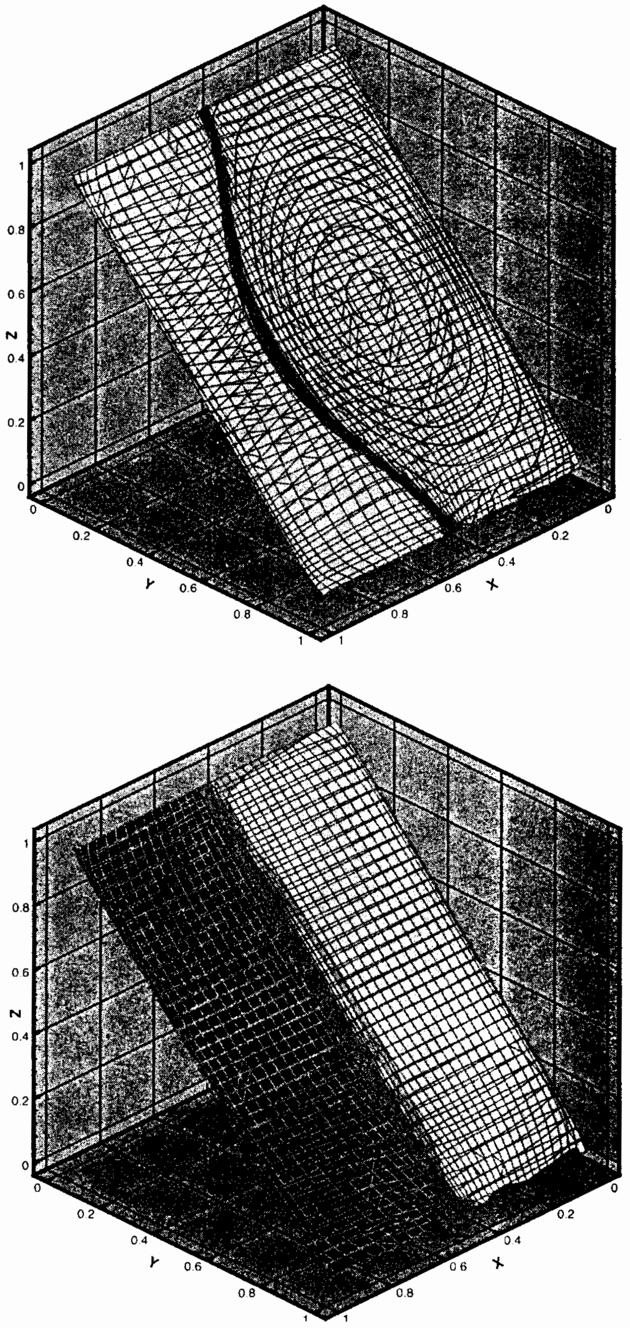

FIGURE 3. Contours of mesh and solution for Example 5.2 on one cut-plane. Top: $t = 0.2$; Bottom: $t = 0.4$.

with time function $g(t) = \cos(\pi t/T)$. The physical domain is considered as a unit cube with periodic boundary conditions at $x = 0, 1$, $y = 0, 1$, and $z = 0, 1$. Initially, the density distribution has a discontinuity placed at $x = \frac{1}{2}$, i.e.

$$\rho(x,y,z,0) = \begin{cases} 1, & \text{if } x \le 0.5, \\ 0, & \text{if } x > 0.5. \end{cases} \tag{5.3}$$

Due to the form of $g(t)$ in the velocity field, the initial discontinuity should be recovered at time $t = T$. Since the initial condition (5.3) is discontinuous, it is reasonable to use the gradient-based monitor function. Therefore, the monitor function is taken as ωI with

$$\omega = \sqrt{1 + 0.5 * |\tilde{\nabla}\rho|^2}, \tag{5.4}$$

where $\tilde{\nabla} := (\partial_\xi,\ \partial_\eta,\ \partial_\zeta)^T$. In the present computations, $T = 0.4$ and 40^3 grid points are used. We also employed the van Leer's limiter, as discussed in [**27**]. Fig. 3 shows results for the adaptive mesh obtained by using the proposed moving mesh method on one cut-plane at $t = 0.5 * T = 0.2$ and $t = T$. It can be verified that the solution at $t = T$ is the same as the initial function. This is well observed in the numerical solutions. It is also observed that our moving mesh scheme adapts the mesh very well to the regions with large solution gradients.

EXAMPLE 5.3. **A spherical Riemann problem between two walls.** The second 3D example is a spherical Riemann problem between two parallel walls at $z = 0$ and $z = 1$. Initially the gas is at rest with density and pressure $(\rho, p) = (1, 1)$ everywhere except in a sphere at $(0, 0, 0.4)$ with radius 0.2. Inside the pressure $(\rho, p) = (1, 5)$. The jump in pressure results in a strong outward moving shock wave and contact discontinuity and an inward rarefaction wave. This inward moving wave causes a local "implosion", and a second outward moving shock wave is created. The main features of the solution are the interactions between these waves and between waves and the walls. This problem is proposed by Langseth and LeVeque [**16**].

Figs. 4 and 5 show the contours of computed density and adaptive meshes at $t = 0.2$, 0.3, 0.5, and 0.6, respectively. In this computation, the following monitor function is used:

$$\omega = \sqrt{1 + 0.5 * |\tilde{\nabla}\rho|^2 + 0.1 * |\tilde{\Delta}\rho|^2}, \tag{5.5}$$

where $\tilde{\Delta} := (\partial_{\xi\xi},\ \partial_{\eta\eta},\ \partial_{\zeta\zeta})^T$. The inclusion of the higher-order derivatives was not used in the two-dimensional moving mesh computations for hyperbolic conservation laws (e.g., [**2, 3, 27**]). However, the monitor (5.5) seems useful for this three-dimensional problem.

6. Concluding Remarks

In this work we proposed an adaptive mesh algorithm for solving hyperbolic system of conservation laws in three space dimensions. The basic idea of our adaptive mesh algorithm can be summarized as the following:

(1) Suppose a logically rectangular spatial grid is given on which the cell centered approximation to the PDE's solution lives.

(2) Update the grid by iterating an artificial time (τ) elliptic (or parabolic) grid generator. Simultaneously update the cell centered approximate solution u by iterating $\partial_\tau u = 0$ on the moving (with respect to τ) grid.

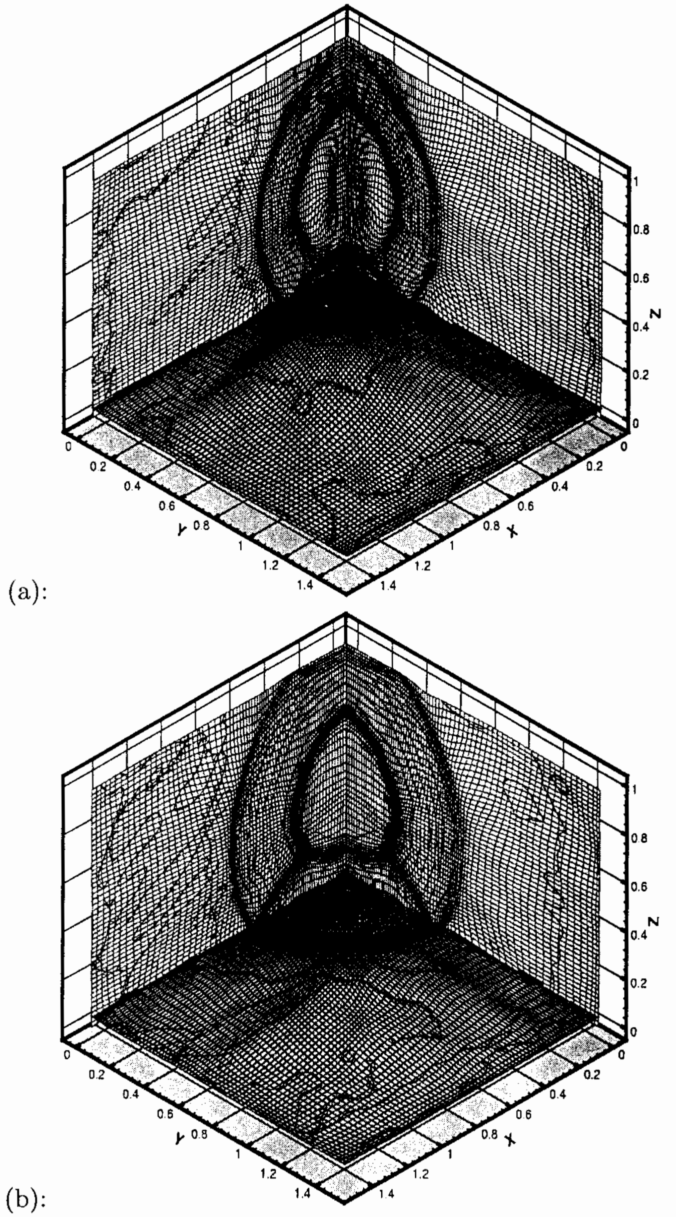

FIGURE 4. Example 5.3: contours of density and mesh at (a): $t = 0.2$, (b) $t = 0.3$, with $75 * 75 * 50$ grid points.

(3) Update u by a physical time Δt according to a fixed-grid conservative difference scheme which is consistent with the original conservation law.

Step 2 above should be regarded as mesh redistribution together with a conservative method to interpolate the approximate solution on the evolving grid. Therefore, our algorithm is in fact a *splitting scheme*: First update the solution to $\partial_t u = 0$

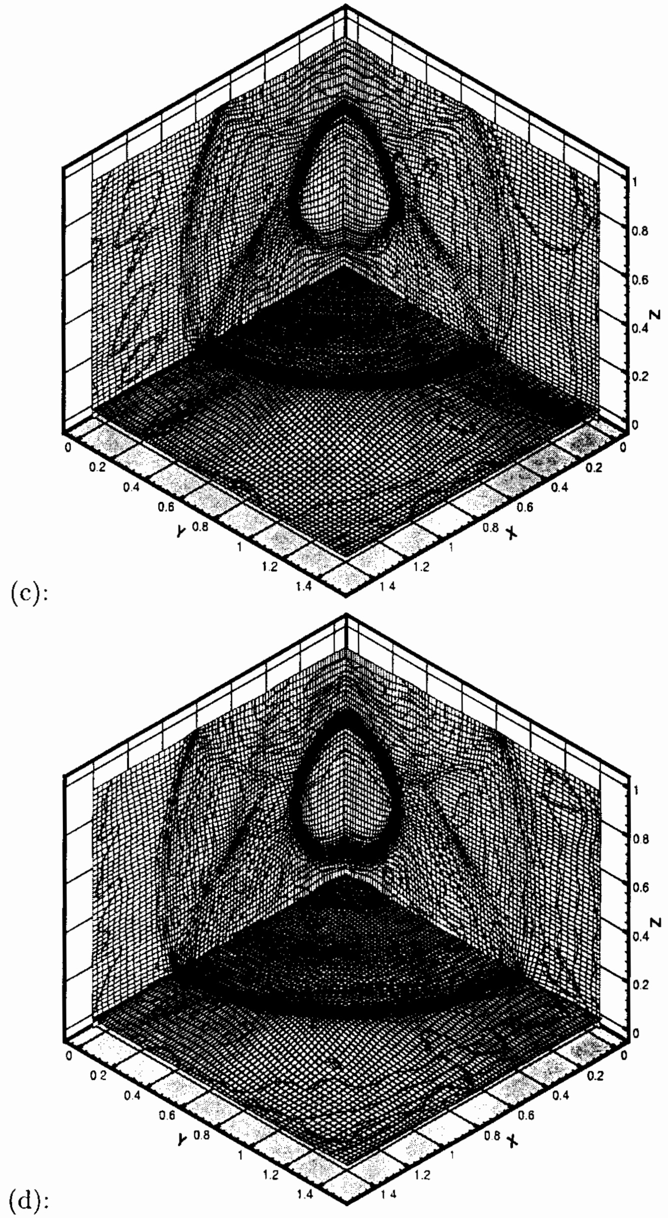

FIGURE 5. Same as Fig. 4, except at (c): $t = 0.5$ and (d): $t = 0.6$.

on a moving grid. Second update this consistent to $\partial_t u + \nabla \cdot f(u) = 0$ on the new fixed grid.

One challenging problem for the moving mesh methods is how to carry out numerical error analysis. Since the mesh redistribution is strongly coupled by the PDE solutions, it is very difficult to establish a theoretical framework to demonstrate the advantage of the moving mesh approach.

Acknowledgment

This work was supported in part by Hong Kong Research Grants Council (CERG HKBU 2044/00P and 2083/01P), the National Natural Science Foundation of China (Grant #19901031) and the Special Funds for Major State Basic Research Projects of China. HZT also wishes to thank the support from Alexander von Humboldt Foundation.

References

[1] B.N. Azarenok, Variational barrier method of adaptive grid generation in hyperbolic problems of gas dynamics. *SIAM J. Numer. Anal.*, 40 (2002), pp. 651-682.

[2] B.N. Azarenok and S.A. Ivanenko, Application of adaptive grids in numerical analysis of time-dependent problems in gas dynamics, *Comput. Maths. Math. Phys.*, 40(2000), pp. 1330-1349.

[3] B.N. Azarenok, S.A. Ivanenko, and T. Tang, Adaptive mesh redistribution method based on Godunov's scheme, *Journal of Communication on Mathematical Sciences*, To appear.

[4] M. Berger and P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* 82(1989), pp.62–84.

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

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

[7] 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.

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

[9] Z. Chen and R. H. Nochetto. Residual type a posteriori error estimates for elliptic obstacle problems. *Numer. Math.*, 84 (2000), pp. 527-548.

[10] E. A. Dorfi and L. O'c. Drury, Simple adaptive grids for 1-D initial value problems, *J. Comput. Phys.*, 69 (1987), pp. 175-195.

[11] M.G. Edwards, J.T. Oden, and L. Demkowicz, An *h*–*r*–adaptive approximate Riemann solver for the Euler equations in two dimensions. *SIAM J. Sci. Comput.* 14 (1993), pp.185-217.

[12] T. Geßner and D. Kröner, Dynamic mesh adaptive for supersonic reactive flow, Preprint, 2000. `http://www.mathematik.univ-freiburg.de/IAM/INDEX.HTM`.

[13] U. Göhner and G. Warnecke: A shock indicator for adaptive transonic flow computations, *Numer. Math.* 66 (1994), pp.423–448.

[14] U. Göhner and G. Warnecke: A second order finite difference error indicator for adaptive transonic flow computations, *Numer. Math.* 70 (1995), pp.129–161.

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

[16] 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.

[17] 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.

[18] R. Li, W.-B. Liu, H.-P. Ma and T. Tang, Adaptive Finite element approximation for distributed elliptic optimal control problems. *SIAM J. Optimization Control*, To appear.

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

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

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

[22] 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.

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

[24] H. Nessyahu and E. Tadmor, Non-oscillatory central differencing for hyperbolic conservation laws, *J. Comput. Phys.* 87 (1990), pp. 408-463.

[25] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes II, *J. Comput. Phys.* 83 (1989), 32-78.

[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] H.Z. Tang and T. Tang, Moving mesh methods for one- and two-dimensional hyperbolic conservation laws, *SIAM J. Numer. Anal.*, To appear.

[28] H.-Z. Tang, T. Tang and P.-W. Zhang, An adaptive mesh redistribution method for nonlinear Hamilton-Jacobian equation in two- and three-dimensions. Submitted to *J. Comput. Phys.* (revised).

[29] H.Z. Tang and H.M. Wu, High resolution KFVS finite volume methods and its applications in CFDs,*Chinese J. Numer. Math. & Appl.* 21(1999), pp.93-103.

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

[31] R. Verfürth, *A Review of a Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, Wiley & Teubner, 1996

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

[33] K. Xu, *Gas-Kinetic Schemes for Unsteady Compressible Flow Simulations*, von Karman Institute for Fluid Dynamics Lecture Series 1998-03 (1998).

School of Mathematical Sciences, Peking University, Beijing 100871, People's Republic of China

E-mail address: hztang@math.pku.edu.cn

Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong and Institute of Computational Mathematics, Chinese Academy of Sciences, Beijing 100080, P.R. China

E-mail address: ttang@hkbu.edu.hk; ttang@lsec.cc.ac.cn