

## An Optimization-Based Rezoning for ALE Methods<sup>†</sup>

Yibing Chen and Song Jiang\*

*Institute of Applied Physics and Computational Mathematics, P.O. Box 8009, Beijing 100088, China.*

Received 25 March 2008; Accepted (in revised version) 30 July 2008

Available online 9 September 2008

---

**Abstract.** Based on the theory of optimization, we use edges and angles of cells to represent the geometric quality of computational grids, employ the local gradients of the flow variables to describe the variation of flow field, and construct a multi-objective programming model. The solution of this optimization problem gives appropriate balance between the geometric quality and adaptation of grids. By solving the optimization problem, we propose a new grid rezoning method, which not only keeps good geometric quality of grids, but also can track rapid changes in the flow field. In particular, it performs well for some complex concave domains with corners. We also incorporate the rezoning method into an Arbitrary Lagrangian-Eulerian (ALE) method which is widely used in the simulation of high-speed multi-material flows. The proposed rezoning and ALE methods of this paper are tested by a number of numerical examples with complex concave domains and compared with some other rezoning methods. The numerical results validate the robustness of the proposed methods.

**AMS subject classifications:** 76M12, 76M10, 76N15, 65K10, 90C29

**Key words:** Grid rezoning, multi-objective programming models, ALE methods.

---

## 1 Introduction

Multi-material flows, where a moving interface exists between two immiscible fluids, can be found in a variety of scientific and engineering problems. Development of numerical accurate and computationally efficient algorithms for multi-material flow simulations remains one of the challenging topics in computational fluid dynamics. Traditionally, numerical methods for multi-material computations have fallen into two classes: Eulerian methods and Lagrangian methods. Eulerian methods hold the mesh of cells fixed and a fluid flows from one cell to another through cell edges via advection. Eulerian methods are robust, capable of running under severe flow conditions (such as under large flow

---

<sup>†</sup>Dedicated to Professor Xiantu He on the occasion of his 70th birthday.

\*Corresponding author. *Email addresses:* chen\_yibing@iapcm.ac.cn (Y. Chen), jiang@iapcm.ac.cn (S. Jiang)

deformation), but may result in badly smeared material interfaces due to numerical diffusion. For Lagrangian methods, cells flow with the fluid and no fluid moves across cell edges. Material interfaces remain intact as they travel with cells. Lagrangian methods are capable of producing sharp interfaces, but may result in mesh contortion and tangling, causing inaccuracy and even breakdown of computation. Over the last decades, another method has been developed that smoothly spans Eulerian and Lagrangian methods offering the benefits of both: the Arbitrary Lagrangian-Eulerian (ALE) method proposed by Hirt, Amsden and Cook (see [25,33]). In the ALE method, the solution algorithm can vary from pure Eulerian to pure Lagrangian through dynamic rezoning and remapping, such that a smooth mesh topology can be maintained, increasing thus accuracy and robustness of the numerical algorithm. A general review of the ALE method can be found in the paper by Benson [9]. There are further developments and applications of this approach, see, e.g., [4, 19, 23, 28, 32, 34, 37].

Generally, an ALE method consists of three phases: The explicit Lagrangian phase, the rezoning phase (mesh movement) and the remapping phase. One of the key factors to a successful ALE method is a robust rezoning algorithm in the rezoning phase that does not require user intervention. In the early development of the ALE methods, the rezoning phase was often carried out by employing a process of grid generation, for which only the geometric quality of the grid was taken into account [1, 17, 20]. However, the geometric quality of a grid is not the only factor that will affect simulation results. From the numerical simulation point of view, it has become a common sense that a good rezoned grid should in general satisfy the following four requirements:

(i) A rezoned grid should remain convex. A lack of control of grid skewness may result in a major deficiency for some algorithms (see, e.g., [2]).

(ii) A rezoned grid should maintain the smoothness, orthogonality and uniformity to increase the computational accuracy. The geometric quality of a grid affects the accuracy of the Lagrangian phase in the ALE methods. In a non-Descartian grid, the numerical error of the Lagrangian phase is not only induced by the truncation error of the used schemes and the grid size, but also depends on the smoothness, orthogonality and uniformity of the grid.

(iii) In the regions where the gradients of the flow variables are large, the distance between the rezoned grid and the old grid must be small in order to keep the remapping error small. Some rezoning methods require that the rezoned grid should be close to the Lagrangian grid, see, e.g., [22, 28, 42], and this idea works well in many cases. But, in some cases other criteria of rezoning are better, see, e.g., [31]. By numerical tests we have found that the distance between the old grid and the rezoned grid plays an important role in resolving local gradients of the flow variables in the remapping phase.

(iv) A rezoned grid should be adaptive to resolve local gradients of the flow variables. Recent progress in the development of  $r$ -adaptive methods (i.e., moving mesh methods) shows that higher accuracy can be achieved by appropriate moving cells to regions of rapid changes in the flow variables, see, e.g., [5, 26, 30, 37, 40].

In view of the above requirements and analysis, we see that a suitable grid movement

is important to ensure both the geometric quality and the grid adaptation.

If only the geometric quality is taken into account, the rezoning phase is essentially equivalent to a process of grid generation. With the development of grid generators, it is now not difficult to find some grid generator which generates a grid satisfying any one of the above four requirements. It is difficult to construct a grid generator that generates a grid satisfying the above four requirements simultaneously.

In two-dimensions, grid generation is in general a process to find a one-to-one mapping  $(x(\xi,\eta),y(\xi,\eta))$  from the physical space  $(x,y)$  to the logical space  $(\xi,\eta)$ . For any included angle in a cell, the following three quantities

$$g_{12} = x_{\xi}x_{\eta} + y_{\xi}y_{\eta}, \quad g_{11} = x_{\xi}^2 + y_{\xi}^2, \quad g_{22} = x_{\eta}^2 + y_{\eta}^2$$

represent the degree of the angle, and the length of the two edges. The variational grid generation, introduced by [41] and further developed in the last decades, is to control following functionals:

$$\text{Smoothness functional:} \quad I = \int_0^1 \int_0^1 \frac{g_{11} + g_{22}}{\det(J)} d\xi d\eta \quad \text{with} \quad J = \begin{bmatrix} x_{\xi} & x_{\eta} \\ y_{\xi} & y_{\eta} \end{bmatrix},$$

$$\text{Length functional:} \quad I = \int_0^1 \int_0^1 (g_{11} + g_{22}) d\xi d\eta,$$

$$\text{Area functional:} \quad I = \int_0^1 \int_0^1 J^2 d\xi d\eta,$$

$$\text{Orthogonality functional:} \quad I = \int_0^1 \int_0^1 g_{12}^2 d\xi d\eta,$$

which measure the smoothness, length, area and orthogonality of grids, respectively. The corresponding Euler-Lagrangian equations of these functionals and their different combinations have led to a number of grid generators, see, e.g., [3, 10–12, 38, 41]. Such grid generators can generate smooth, uniform and orthogonal grids in many cases, and they work well in regular domains. However, as observed in [15, 22, 29, 42], they often fail in complex domains because the grids generated by these methods may become folded in the vicinity of a corner. To circumvent this drawback, Charakhch'yan and Ivanenko proposed a generalized Winslow (GWinslow) method [15], which could generate a convex and smooth grid for many complex domains. We should point out here that in the methods given in [3, 15, 38, 41], only the geometric quality of grids has been addressed, but the orthogonality is not taken into account. To reach higher accuracy for the ALE methods, Knupp et al. have recently proposed the the reference Jacobian matrix (RJM) method [28]; the basic idea of which is to ensure the geometric quality of a rezoned grid, while keeping the rezoned grid as close as possible to the old grid. This method and some other similar methods [22, 42] perform well and have been applied to the ALE codes. On the other hand, since the rezoned grid has to be close to the Lagrangian grid, the constraint conditions for grid rezoning in [22, 28, 42] seem too strict, so that the adaptation of grids seems difficult to be considered simultaneously.

In the past decades, various adaptive methods have been developed and successfully applied to a wide range of fluid dynamical problems. Among adaptive methods, moving mesh methods (i.e.,  $r$ -adaptive methods) have attracted more and more attention. The basic idea of moving mesh methods is that a fixed number of nodes with fixed connectivity is moved within the computational domain to resolve local gradients of the flow variables and to increase numerical accuracy. A moving mesh method can be used as an independent module to an existing code without any change of the data structure, making it practical in applications. We refer the reader to the paper [24] for a review on the early developments, and the papers [13, 14, 30] and the references cited therein on the recent developments for moving finite difference and finite element methods. Most of these studies concentrate on unstructured triangular meshes. Based on different strategies, several moving mesh methods on structured meshes have been studied recently [5–7, 39, 40] which are of great interest in applications, since there are many applied CFD codes which still use structured meshes. The moving mesh method proposed in [39, 40] is easy to code and work quite well in regular computational domains, but may fail in complex domains (see the numerical examples in Section 5). On the other hand, the method of Azarenok et al. [5, 7] can handle complex domains, but does not satisfy the aforementioned four requirements since their method is extended from Charakhch'yan and Ivanenko's generalized Winslow method and possesses the smoothness property of grids only. The methods in [5–7, 39, 40] have shown success for some fluid problems in the Eulerian framework. In the ALE framework, a good grid may contort and tangle in the next time step in the Lagrangian phase, so the mesh movement is more subtle. Jin and Xu [27] combined the mesh redistribution used in Tang and Tang [39, 40] with Hui's unified coordinates method [21] to simulate dynamics of a freely falling plate, and obtained good numerical results in comparison with the experiment. However, a problem similar to the mesh redistribution used in Tang and Tang [40] will appear when extending their simulation to more complex domains. In [31], Lipnikov and Shashkov presented a one-dimensional moving mesh method for the ALE codes, which is shown numerically to be superior to many existing rezoning methods.

Inspired by the work of Lipnikov and Shashkov [31], in this paper we will propose a new two-dimensional rezoning (mesh movement) method, which is not based on the traditional variational method but on the theory of optimization. Particular attention is paid to satisfy the aforementioned four requirements. Consequently, based on our rezoning method, a new moving mesh method for the ALE codes is obtained. We have tested our method by a number of numerical examples, which demonstrate that the proposed method performs well for both regular and complex computational domains.

The outline of this paper is as follows. In the next section we briefly describe an ALE method to be used in our rezoning method. In Section 3, we present our optimization-based rezoning (mesh movement) method. In Section 4 we test our method by a number of numerical examples, and compare it with the Winslow and GWinslow methods as well as the mesh redistribution used by Tang and Tang [40]. The proposed rezoning method is also tested under the ALE framework described in Section 2.

## 2 An ALE method for compressible flows

The aim of this paper is to give a new rezoning method in the rezoning phase of the ALE methods. For completeness we recall in brief an ALE method for the two-dimensional compressible Euler equations [16], in which the physical quantities such as the density, pressure and energy are defined at the cell center, while the velocity is defined at the cell nodes. We will incorporate our rezoning method in this ALE method to give three numerical tests for the Euler equations.

The two-dimensional Euler equations for the ALE methods can be written as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{D}) = -\nabla \cdot (\rho \vec{w}), \quad (2.1)$$

$$\rho \left( \frac{\partial v_x}{\partial t} + \vec{D} \cdot \nabla v_x \right) = -\frac{\partial p}{\partial x} - \rho \vec{w} \cdot \nabla v_x, \quad (2.2)$$

$$\rho \left( \frac{\partial v_y}{\partial t} + \vec{D} \cdot \nabla v_y \right) = -\frac{\partial p}{\partial y} - \rho \vec{w} \cdot \nabla v_y, \quad (2.3)$$

$$\frac{\partial(\rho e)}{\partial t} + \nabla \cdot (\rho e \vec{D}) = -p \nabla \cdot \vec{v} - \nabla \cdot (\rho e \vec{w}) \quad (2.4)$$

with the equation of state

$$p = (\gamma - 1) \rho \left( e - \frac{1}{2} |\vec{v}|^2 \right), \quad (2.5)$$

where  $\vec{w} = \vec{v} - \vec{D} \equiv (w_x, w_y)$ , and  $\rho$ ,  $e$ ,  $\vec{v} = (v_x, v_y)$ ,  $p$ , and  $\vec{D}$  are the density, the specific internal energy, and the fluid velocity, the pressure, and the mesh velocity, respectively.

Next, we describe the discretization of (2.1)-(2.4). In view of great success of the DYNA code by Hallquist [20], we will use here the same discretization as that in DYNA, i.e., the finite volume discretization for the mass and energy equations, together with the finite element discretization for the momentum equation. Such discretization for (2.1)-(2.4) is widely used in the simulation of high-speed multi-material flows.

### 2.1 Mass equation

First, we use a finite volume method to discretize the mass equation (2.1). We integrate (2.1) over a cell  $\Omega(t)$  (see Fig. 1(a)) to deduce that

$$\int_{\Omega(t)} \left( \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{D}) \right) dV = - \int_{\Omega(t)} \nabla \cdot (\rho \vec{w}) dV. \quad (2.6)$$

Here the left hand side of (2.6) at  $t = t^n$  is approximated as follows:

$$\int_{\Omega(t^n)} \left( \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{D}) \right) dV = \frac{dM}{dt} \Big|_{t=t^n} \approx \frac{M^{n+1} - M^n}{\Delta t}, \quad (2.7)$$

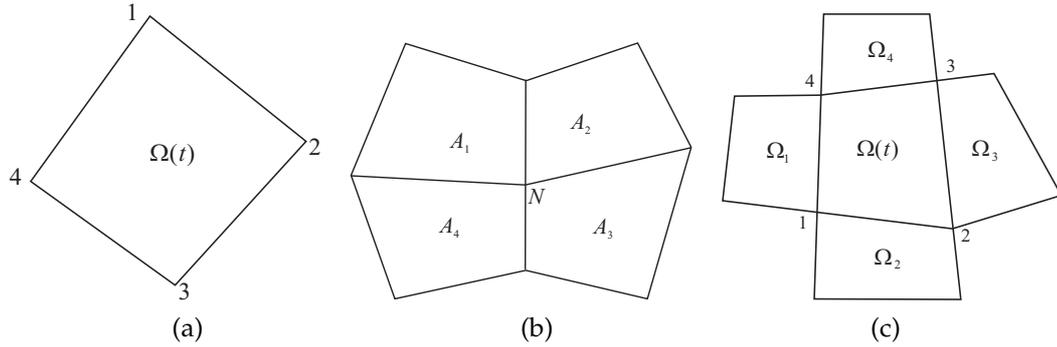


Figure 1: (a) Cell  $\Omega(t)$  with four nodes. (b): Four cells around a node. (c). Four neighboring cells of  $\Omega(t)$ .

where  $M = \int_{\Omega(t)} \rho dV$  is the mass in  $\Omega(t)$  and  $M^\ell = \int_{\Omega(t^\ell)} \rho dV$  ( $\ell = n, n+1$ ). Moreover, the right-hand side is discretized as follows:

$$\int_{\Omega(t^n)} \nabla \cdot (\rho \vec{w}) dV = \int_{\Omega(t^n)} (\rho \vec{w}) \cdot \vec{n} dV \approx \sum_{i=1}^4 \rho_{i,i+1}^n (w_y^n \Delta x - w_x^n \Delta y)_{i,i+1}, \quad (2.8)$$

where  $\vec{n}$  is the outer normal vector,  $i$  is the node number of the cell  $\Omega(t^n)$ ,

$$(w_y^n)_{i,i+1} = \frac{(w_y^n)_i + (w_y^n)_{i+1}}{2}, \quad (w_x^n)_{i,i+1} = \frac{(w_x^n)_i + (w_x^n)_{i+1}}{2},$$

$(w_x^n)_i$  is the value of  $w_x$  at the node  $i$  and time  $t^n$ , and  $\rho_{i,i+1}^n$  is the value of the density at the center of the boundary segment  $(i, i+1)$ .

We can apply the second-order MUSCL scheme to calculate  $\rho_{i,i+1}^n$  in our simulations [8]. Without loss of generality, we discuss how to obtain  $\rho_{4,1}$  only (see Fig. 1(c)), and the other  $\rho_{i,i+1}^n$  can be calculated in a similar manner. Denote by  $\rho_1, \rho_\Omega$  and  $\rho_3$  the density in  $\Omega_1, \Omega(t)$  and  $\Omega_3$ , and by  $V_1, V_\Omega$  and  $V_3$  the volume of  $\Omega_1, \Omega(t)$  and  $\Omega_3$  respectively, see Fig. 1(c). We now define the discrete derivative of  $\rho$  by

$$\left( \frac{\partial \rho}{\partial V} \right)_{\Omega(t)} := \frac{1}{2} \left( 1 + \text{sign}(s_a \cdot s_b) \right) \frac{s_a \cdot s_b}{s_a + s_b},$$

where

$$s_a = \frac{\rho_1 - \rho_\Omega}{\frac{1}{2}(V_1 + V_\Omega)}, \quad s_b = \frac{-\rho_3 + \rho_\Omega}{\frac{1}{2}(V_3 + V_\Omega)}.$$

Thus, the piecewise linear reconstruction of the density at the boundary segment  $l_{4,1}$  and  $l_{3,2}$  of  $\Omega(t)$  is given by

$$\begin{aligned} \tilde{\rho}_{\Omega(t), l_{4,1}} &:= \rho_{\Omega(t), l_{4,1}} - \frac{1}{2} \left( \frac{\partial \rho}{\partial V} \right)_{\Omega(t)} \cdot V_{\Omega(t)}, \\ \tilde{\rho}_{\Omega(t), l_{3,2}} &:= \rho_{\Omega(t), l_{3,2}} + \frac{1}{2} \left( \frac{\partial \rho}{\partial V} \right)_{\Omega(t)} \cdot V_{\Omega(t)}. \end{aligned}$$

In the same way, we can obtain  $\tilde{\rho}_{\Omega_1, l_{4,1}}$ . Finally, we take  $\rho_{4,1}$  to be

$$\rho_{4,1} := \begin{cases} \tilde{\rho}_{\Omega(t), l_{4,1}}, & \text{if } \vec{w}_{4,1} \cdot \vec{n}_{4,1} > 0, \\ \tilde{\rho}_{\Omega_1, l_{4,1}}, & \text{otherwise,} \end{cases}$$

where  $\vec{n}_{4,1}$  is the outer normal vector of the edge  $l_{4,1}$ .

Inserting (2.7) and (2.8) into (2.6), we obtain the discretization form of (2.1), where the unknown is the mass in a cell:

$$M^{n+1} = M^n - \Delta t \sum_{i=1}^4 \rho_{i,i+1}^n (w_y^n \Delta x - w_x^n \Delta y)_{i,i+1}. \tag{2.9}$$

### 2.2 Energy equation

We still use a finite volume method to discretize the energy equation. Integrating (2.4) over a cell  $\Omega(t)$ , we have

$$\int_{\Omega(t)} \left( \frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho e \vec{D}) \right) dV = - \int_{\Omega(t)} \nabla \cdot (\rho e \vec{w}) dV - \int_{\Omega(t)} p \nabla \cdot \vec{v} dV. \tag{2.10}$$

Here the left hand side and the last term on the right hand side at time  $t^n$  are approximated as follows:

$$\int_{\Omega(t^n)} \left( \frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho e \vec{D}) \right) dV = \frac{dE}{dt} \Big|_{t=t^n} \approx \frac{E^{n+1} - E^n}{\Delta t} \tag{2.11}$$

with  $E = \int_{\Omega(t)} \rho e dV \approx e(t) \int_{\Omega(t)} \rho dV = e(t) M(t)$  being the internal energy in  $\Omega(t)$ , and

$$- \int_{\Omega(t^n)} p \nabla \cdot \vec{v} dV \approx - p^{n+1/2} \int_{\Omega(t^n)} \nabla \cdot \vec{v} dV \approx - \frac{(p^n + p^{n+1}) \Delta V^{n+1/2}}{2 \Delta t}, \tag{2.12}$$

where  $\Delta V^{n+1/2} = V(\Omega_L(t^{n+1})) - V(\Omega(t^n))$ ,  $\Omega_L(t^{n+1})$  is the quadrilateral formed by the vertices  $\vec{r}_L^{n+1} = \vec{r}^n + \Delta t \vec{v}^n$ ,  $\vec{r}^n$  are the vertices of the quadrilateral  $\Omega(t^n)$ . Throughout this section  $V(G)$  denotes the volume of  $G$ .

The discrete form for the first term on the right hand side of (2.10)

$$g_e(t^n) := - \int_{\Omega(t^n)} \nabla \cdot (\rho e \vec{w}) dV$$

can be obtained in the same manner as that for  $\int_{\Omega(t^n)} \nabla \cdot (\rho \vec{w}) dV$ . Substituting (2.11) and (2.12) into (2.10), we obtain the discrete form of (2.4), in which the unknown is the internal energy defined at the center of the cell:

$$\frac{E^{n+1} - E^n}{\Delta t} = - \left( \frac{p^n + p^{n+1}}{2} + q^{n+1/2} \right) \frac{\Delta V_L^{n+1}}{\Delta t} + g_e(t^n), \tag{2.13}$$

where we have added some artificial viscosity  $q^{n+1/2}$  to eliminate possible numerical oscillations across strong discontinuities, and in our simulations  $q^{n+1/2}$  is chosen to be the Von Neumann and Richtmyer viscosity [9]:

$$q^{n+1/2} = \begin{cases} 0, & \text{if } \nabla \cdot \bar{v}^n \geq 0, \\ \rho^{n+1/2} l^{n+1/2} |\nabla \cdot \bar{v}^n| (q_1 l^{n+1/2} |\nabla \cdot \bar{v}^n| + q_2 c^{n+1}), & \text{otherwise,} \end{cases} \quad (2.14)$$

where  $q_1$  and  $q_2$  are constant parameters, which are taken as  $q_1 = 1.5$  and  $q_2 = 0.6$  in our simulations;  $l^{n+1/2} = \sqrt{2} S^{n+1/2} / l_{\max}^{n+1/2}$ , and  $S^{n+1/2}$  is the area of the cell at time  $t^{n+1/2}$ ,  $l_{\max}^{n+1/2}$  is the maximal length of diagonals of the cell,  $c^{n+1}$  is the speed of sound at time  $t^{n+1}$ .

### 2.3 Momentum equations

We use a finite element method to discretize the momentum equations (2.2) and (2.3). The corresponding finite element space is given by

$$U_h = \{v_h | v_h \in C(\cup_k \bar{\Omega}_k), v_h \text{ is a bilinear function in every } \Omega_k\}.$$

On the reference element  $\{-1 < \xi, \eta < 1\}$  in the reference space  $(\xi, \eta)$ , we define the reference shape functions:

$$\psi_i(\xi, \eta) = \frac{1}{4} (1 + \xi \xi_i) (1 + \eta \eta_i), \quad i = 1, 2, 3, 4,$$

where  $(\xi_1, \xi_2, \xi_3, \xi_4) = (-1, 1, 1, -1)$  and  $(\eta_1, \eta_2, \eta_3, \eta_4) = (-1, -1, 1, 1)$ .

Let  $(x_i, y_i)$  ( $i=1, 2, 3, 4$ ) denote the coordinates of the four vertices of a quadrilateral element  $\Omega(t)$  in the space  $(x, y)$  (see Fig. 1(a)). The element  $\Omega(t)$  is projected to the reference space  $(\xi, \eta)$  by the following coordinate transform, denoted by  $F$ ,

$$x = \sum_{i=1}^4 x_i \psi_i(\xi, \eta), \quad y = \sum_{i=1}^4 y_i \psi_i(\xi, \eta).$$

We define

$$\psi_i(\xi, \eta) = \psi_i(\xi(x, y), \eta(x, y)) =: \phi_i(x, y),$$

and choose  $\phi_i(x, y)$  ( $i=1, 2, 3, 4$ ) as the basis of the finite element space.

Let  $N$  be an inner node, and  $A_1, \dots, A_4$  the surrounding elements around  $N$  (see Fig. 1(b)). Each element  $A_j$  ( $1 \leq j \leq 4$ ) is projected onto the reference element in the space  $(\xi, \eta)$  by the transform  $F_j$ . We define the test function  $\varphi_N$  corresponding to the node  $N$  by

$$\varphi_N = \begin{cases} \phi_{i,j}, & \text{if } (x, y) \in A_j \text{ and the vertex } N \text{ is the } i^{\text{th}} \text{ node of } A_j, 1 \leq j \leq 4, \\ 0, & \text{otherwise.} \end{cases}$$

We now multiply Eq. (2.3) by  $\varphi_N$  and integrate the resulting equation over  $A(t) := \cup_{j=1}^4 A_j \equiv \cup_{j=1}^4 A_j(t)$  to deduce

$$\int_{A(t)} \rho \left( \frac{\partial v_x}{\partial t} + \vec{D} \cdot \nabla v_x \right) \varphi_N dx dy = - \int_{A(t)} \frac{\partial p}{\partial x} \varphi_N dx dy - \int_{A(t)} \rho \vec{w} \cdot \nabla v_x \varphi_N dx dy, \quad (2.15)$$

where all the terms in (2.15) are discretized as follows.

$$\int_{A(t^n)} \rho \left( \frac{\partial v_x}{\partial t} + \vec{D} \cdot \nabla v_x \right) \varphi_N dx dy \approx \frac{v_{x,N}^{n+1} - v_{x,N}^n}{\Delta t} \sum_{k=1}^4 \frac{1}{4} M_{k,A}^{n+\frac{1}{2}}, \quad (2.16)$$

where  $v_{x,i}$  is the value of  $v_x$  at the node  $i$ , and  $M_{k,A}^{n+\frac{1}{2}}$  is the mass in the cell  $A_k$  around the node  $N$  (cf. Fig. 1(b));

$$\int_{A(t^n)} \frac{\partial p}{\partial x} \varphi_N dx dy = - \int_{A(t^n)} p \frac{\partial \varphi_N}{\partial x} dx dy = - \sum_{i=1}^4 (p_i^{n+\frac{1}{2}} + q_i^{n+\frac{1}{2}}) \frac{y_{i,2}^{n+\frac{1}{2}} - y_{i,4}^{n+\frac{1}{2}}}{2}, \quad (2.17)$$

where  $q_i^{n+\frac{1}{2}}$  is the artificial viscosity given by (2.14);

$$\begin{aligned} & \int_{A(t^n)} \rho \vec{w} \cdot (\nabla v_x) \varphi_N dx dy \\ &= \sum_{i=1}^4 \frac{\rho_i^{n+\frac{1}{2}}}{2} (v_{x,N}^n - \bar{v}_{x,i}^n) \left[ (\bar{w}_y)_i (x_{i,2}^{n+\frac{1}{2}} - x_{i,4}^{n+\frac{1}{2}}) - (\bar{w}_x)_i (y_{i,2}^{n+\frac{1}{2}} - y_{i,4}^{n+\frac{1}{2}}) \right], \end{aligned} \quad (2.18)$$

where  $(x_{i,k}^{n+\frac{1}{2}}, y_{i,k}^{n+\frac{1}{2}})$  is the coordinate  $(x, y)$  of the node  $k$  in the cell  $i$  at time  $t^{n+\frac{1}{2}}$ ,

$$(\bar{w}_{y,x})_i = \frac{1}{4} \sum_{k=1}^4 (w_{y,x}^n)_{i,k}, \quad (\bar{v}_x)_i = \frac{1}{4} \sum_{k=1}^4 (v_x^n)_{i,k}$$

$(w_y)_{i,k}, (v_x)_{i,k}$  are the value of  $w_y$  and  $v_x$  at the node  $k$  of the cell  $i$  and at time  $t^{n+\frac{1}{2}}$ , respectively.

Substituting (2.16)-(2.18) into (2.15), we obtain the discretization form for  $v_{x,N}^{n+1}$  in which the unknown is the value of the  $x$ -velocity  $v_x$  at the node  $N$ . In the same manner, we can obtain the discretization form for  $v_{y,N}^{n+1}$ .

### 2.4 Solution procedure

In our moving mesh ALE formulation, the time evolution stage from  $t^n$  to  $t^{n+1}$  consists of three steps:

- (1) The grid and density at time  $t^{n+1}$  are obtained by an explicit Lagrangian step as follows:

$$\begin{aligned}\vec{r}^{n+1} &= \vec{r}^n + \Delta t \cdot \vec{v}^n, & \vec{r} &= (x, y); \\ \rho_j^{n+1} &= \rho_j^n V(\Omega_j^n) / V(\Omega_j^{n+1}).\end{aligned}$$

- (2) The new grid at time  $t^{n+1}$  is updated by our rezoning method which will be introduced below. Thus, the mesh velocity  $\vec{D}$  is obtained.
- (3) Solve numerically Eqs. (2.1)-(2.4) to update all the physical quantities by the ALE method described above (see [16] for more details).

### 3 An optimization-based rezoning method

For the sake of presentation we first describe the idea of our rezoning (mesh movement) method in the case that only the geometric quality of grids is taken into account. Then, in Subsection 3.2 we extend our method to the general case where both the geometric quality and the adaptation are taken into account.

#### 3.1 Based on the geometric quality only

Roughly speaking, our idea is to build a new framework to describe the geometric quality, and to show then that the adaptation can be easily incorporated into this framework. For this purpose, we require that a rezoned grid should possess the properties: 1. Unfoldness/convexity, 2. Orthogonality, 3. Smoothness and 4. Uniformity [29].

Let  $\Omega(t)$  be a cell in a structured quadrilateral grid with four included angles and four edges (cf. Fig. 1(a)). We will use included angles and edges of the grid to describe the above mentioned four properties. To this end, first, given the direction (clockwise or anticlockwise), we require that four included angles of the cell  $\Omega(t)$  should be smaller than  $\pi$ . This guarantees the convexity of the grid. Second, to maintain the orthogonality, we require that all included angles of the grid with the common vertex should be as close in size to each other as possible. In fact, the orthogonality always implies the smoothness of the grid. Finally, In order to assure the uniformity, we require that all edges of the grid conjunct to the same vertex should be as close in length to each other as possible. Next, we discuss how to rezoning the grid according these principles. We begin with the simple case.

##### 3.1.1 Movement of a single node

Consider four cells with nine nodes  $A, B, C, D, E, F, G, H, P$ , see Fig. 2. Corresponding to these four cells, there are sixteen angles and twelve edges. Assume that the node  $P$  can

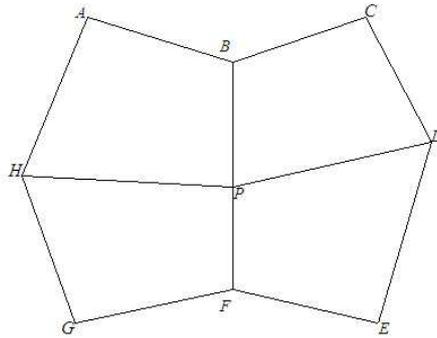


Figure 2: Octagon  $ABCDEFGH$  consists of four cells.

move within the octagon  $ABCDEFGH$ . Obviously, twelve of the sixteen angles and the four edges meeting at the node  $P$  vary with movement of  $P$ .

To describe the convexity of the grid, we require that all twelve angles that vary with movement of  $P$  should be smaller than  $\pi$ . Thus, we define that the set  $\mathcal{S} \subset \mathbb{R}^2$  is feasible, if for any node  $P \in \mathcal{S}$ , all twelve angles  $\angle ABP, \angle PBC, \angle BCD, \angle CDP, \angle PDE, \angle DEF, \angle EFP, \angle PFG, \angle FGH, \angle HGP, \angle PHA$ , and  $\angle HAB$  are smaller than  $\pi$ . Evidently, a convex grid is obtained if  $P$  moves within  $\mathcal{S}$ . Hence, we can easily draw the following conclusion.

**Proposition 3.1.** For a given grid, if there exists a feasible set  $\mathcal{S}$ , then  $\mathcal{S}$  is convex.

The proof of Proposition 3.1 will be given in Appendix. Next, to assure the orthogonality, smoothness and uniformity of the grid, we propose the following two criteria:

1. All angles with the common vertex should be as close in size to each other as possible;
2. All edges meeting at a node should be as close in length to each other as possible.

In view of these criteria, we define the angle function  $f_{\alpha B}$  corresponding to the node  $B$ , which controls the variation of the angles  $\angle ABP$  and  $\angle CBP$ , by

$$f_{\alpha B} := \frac{\angle ABP^2 + \angle CBP^2}{\angle ABP \cdot \angle CBP}.$$

The value of  $f_{\alpha B}$  varies in the interval  $[2, \infty]$  (cf. Fig. 3), and  $f_{\alpha B}$  reaches its minimum when two angles  $\angle ABP$  and  $\angle CBP$  are equal. In the same manner, we can define the angle functions  $f_{\alpha D}$ ,  $f_{\alpha F}$  and  $f_{\alpha H}$  corresponding to the nodes  $D$ ,  $F$  and  $H$ , respectively. Since the movement of the node  $P$  can affect all angles with vertex  $P$ , we define the angle function  $f_{\alpha P}$  corresponding to  $P$  differently. Denoting

$$\begin{aligned} \min P_{\alpha} &= \min\{\angle BPH, \angle BPD, \angle DPF, \angle FPH\}, \\ \max P_{\alpha} &= \max\{\angle BPH, \angle BPD, \angle DPF, \angle FPH\}, \end{aligned}$$

we define

$$f_{\alpha P} := \frac{(\min P_{\alpha})^2 + (\max P_{\alpha})^2}{\min P_{\alpha} \cdot \max P_{\alpha}}.$$

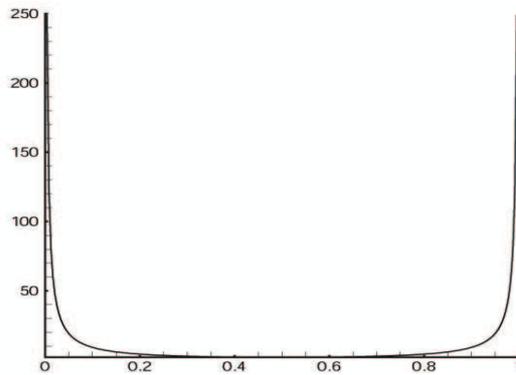


Figure 3: Range of values of the angle function  $f_{\alpha B}$ .

The function  $f_{\alpha P}$  reaches its minimum when all the angles with vertex  $P$  are equal. Finally, we define the angle function  $f_{\alpha}^P$  by

$$f_{\alpha}^P = f_{\alpha B} + f_{\alpha D} + f_{\alpha F} + f_{\alpha H} + f_{\alpha P}.$$

To control the variation of four edges meeting at the node  $P$ , we define the length function  $f_l^P$  by

$$f_l^P := \sqrt{\left(\frac{|BP|^2 + |FP|^2}{|BP| \cdot |FP|}\right)^2 + \left(\frac{|DP|^2 + |HP|^2}{|DP| \cdot |HP|}\right)^2},$$

which reaches its minimum when all the edges have the same length.

Thus, our rezoning method is to move the node  $P$  in the feasible set  $\mathcal{S}$ , such that the following multi-objective programming model is solvable.

$$J = \min_{P \in \mathcal{S}} \{f_{\alpha}^P, f_l^P\}, \text{ such that the angles } \in (0, \pi), \tag{3.1}$$

where the angles mean all angles that affect the movement of the node  $P$ . A non-inferior solution of the problem (3.1) gives a suitable balance among the orthogonality, smoothness and uniformity, while the constraint condition (i.e., the angles  $\in (0, \pi)$ ) guarantees the convexity of the grid. However, there may have infinite many non-inferior solutions to the problem (3.1). To circumvent this shortcoming, instead of using the multi-objective programming model (3.1), we consider the following single-objective programming model:

$$J = \min_{P \in \mathcal{S}} (f_{\alpha}^P \cdot f_l^P), \text{ and the angles } \in (0, \pi), \tag{3.2}$$

where  $\mathcal{S}$  is the feasible set of the octagon  $ABCDEFGH$  and the angles mean all angles that affect the movement of the node  $P$ . We have the following proposition.

**Proposition 3.2.** There exists an optimal solution of the problem (3.2) which is also a non-inferior solution to the problem (3.1).

We will prove Proposition 3.2 in Appendix. By Proposition 3.2, we only need to solve the problem (3.2) whose solution controls the grid movement. Because the feasible set  $\mathcal{S}$  is convex, the problem (3.2) can be considered as an unconstrained optimal programming model in the interior of the feasible set  $\mathcal{S}$ . Thus, the classical numerical methods, such as the steepest decent method and the conjugate gradient method, can be used to solve (3.2) numerically. The algorithm of the steepest decent method can be briefly described by the following loop.

- (a) Given the initial data  $P^{(1)} \in \mathcal{S}$  and the iteration error  $\varepsilon > 0$ , set  $k = 1$ .  
 (b) Compute the search direction  $d^{(k)} = -\nabla(f_\alpha^P \cdot f_l^P)(P^{(k)})$ .  
 (c) If  $\|d^{(d)}\| < \varepsilon$ , then stop the loop. Otherwise, use the line search method to find  $\lambda_k$  by solving

$$f_\alpha^{P^{(k)}} f_l^{P^{(k)}}(P^{(k)} + \lambda_k d^{(k)}) = \min_{\lambda \geq 0, P^{(k)} + \lambda_k d^{(k)} \in \mathcal{S}} f_\alpha^{P^{(k)}} f_l^{P^{(k)}}(P^{(k)} + \lambda_k d^{(k)}).$$

Then, let  $P^{(k+1)} = P^{(k)} + \lambda_k d^{(k)}$  and  $k := k + 1$ , goto the step (b).

### 3.1.2 General case

The rezoning method by a suitable movement of a single node described in the above subsection can be easily extended to all nodes.

Let  $P_i, i = 1, \dots, N$ , denote the interior nodes of the grid, and  $f_{\alpha P_i}$  and  $f_{l P_i}$  the angle and length functions corresponding to the node  $P_i$ , respectively. We construct the following multi-objective programming model to control movement of all the interior nodes:

$$J = \min_{\{P_1, P_2, \dots, P_N\}} \{ \{f_{\alpha}^{P_1}, f_l^{P_1}\}, \dots, \{f_{\alpha}^{P_N}, f_l^{P_N}\} \}, \text{ such that the angles } \in (0, \pi), \quad (3.3)$$

where the angles mean all included angles of the grid. Similar to (3.2), we can also deduce a single-objective programming model corresponding to (3.3) as follows.

$$J = \min_{\{P_1, P_2, \dots, P_N\}} \sum_{i=1}^N (f_{\alpha}^{P_i} \cdot f_l^{P_i})^2, \text{ such that the angles } \in (0, \pi). \quad (3.4)$$

Analogous to Proposition 3.2, we can also show the following proposition, the proof of which will be given in Appendix.

**Proposition 3.3.** There exists an optimal solution of the problem (3.4) which is also a non-inferior solution of the problem (3.3).

The algorithm for the above optimal problem can be described by the following loop.

(a) Given the iteration error  $\varepsilon > 0$ , set  $k = 1$ .

(b) Let  $k := k + 1$  and  $Q_i = P_i^{(k)}$  ( $1 \leq i \leq N$ ). For  $i = 1, \dots, N$ , solve the optimal problem

$$J = \min_{\{Q_i\}} \sum_{i=1}^N (f_\alpha^{Q_i} \cdot f_l^{Q_i})^2, \text{ such that the angles } \in (0, \pi). \quad (3.5)$$

The algorithm for solving (3.5) is the same as the loop described at the end of Section 3.1.1.

(c) Let  $P_i^{(k)} = Q_i$ ,  $i = 1, \dots, N$ . If  $(\sum_{i=1}^N (P_i^{(k+1)} - P_i^{(k)})^2)^{1/2} < \varepsilon$ , then stop the loop. Otherwise, goto (b).

### 3.2 Based on both the geometric quality and the adaptation

In Subsection 3.1 we have presented our rezoning method by controlling only the geometric quality of grids, namely, angles and the length of edges of cells. Actually, the adaptation of grids according to the variation of flow field can be easily incorporated into our method.

Therefore, in this subsection we combine our method with adaptation to present a rezoning method which takes into account both the geometric quality and the adaptation. Our idea is to weight the length of each edge with a weighted monitor function, which reflects the variation of the flow variables. We begin with the simple case of a single node to describe the idea. To rezone a grid by controlling movement of a single node  $P$ , we modify the multi-objective programming model (3.1) as follows:

$$J = \min_P \{f_\alpha^P, f_{l\rho}^P\}, \text{ such that the angles } \in (0, \pi), \quad (3.6)$$

where  $\rho$  is the density of the fluid, and

$$f_{l\rho}^P = \sqrt{\left(\frac{|BP_\rho|^2 + |FP_\rho|^2}{|BP_\rho| \cdot |FP_\rho|}\right)^2 + \left(\frac{|DP_\rho|^2 + |HP_\rho|^2}{|DP_\rho| \cdot |HP_\rho|}\right)^2},$$

$BP_\rho = BP \cdot f_\rho$ , the weighted function  $f_\rho$  represents the variation of flow field. The choice of  $f_\rho$  depends on problems under consideration. For our numerical examples in Section 4 we take

$$f_\rho = \sqrt{1 + a\rho^2}, \quad \text{or} \quad f_\rho = \sqrt{1 + a\rho_\xi^2 + b\rho_\eta^2}, \quad (3.7)$$

where  $a$  and  $b$  are constant parameters. To find a solution of (3.6), as discussed in Subsection 3.1, we need only to find an optimal solution of the following single-objective programming model:

$$J = \min_P f_\alpha^P \cdot f_{l\rho}^P, \text{ such that the angles } \in (0, \pi). \quad (3.8)$$

The above approach can be easily extended to the general case by modifying the optimization problem (3.4) as follows:

$$J = \min_{\{P_1, P_2, \dots, P_N\}} \sum_{i=1}^N (f_{\alpha}^{P_i} \cdot f_{l\rho}^{P_i})^2, \text{ such that the angles } \in (0, \pi). \quad (3.9)$$

In the one-dimensional case, this choice represents the equi-distribution principle. In fact, in this case we only need to consider the length function

$$J = \min_{\{P_1, P_2, \dots, P_N\}} \sum_{i=1}^N (f_{l\rho}^{P_i})^2, \quad (3.10)$$

where

$$f_{l\rho}^{P_i} = \frac{|P_i P_{i+1} \cdot f_{i,i+1,\rho}|^2 + |P_{i-1} P_i \cdot f_{i-1,i,\rho}|^2}{|P_i P_{i+1} \cdot f_{i,i+1,\rho}| |P_{i-1} P_i \cdot f_{i-1,i,\rho}|}.$$

By solving (3.10), we can get the traditional equi-distribution principle, namely,

$$|P_i P_{i+1}| \cdot f_{i,i+1,\rho} = \text{constant}.$$

In summary, our rezoning method is to move the grid in such a way that the single-objective programming model (3.9) is solvable. In the next section we will test this rezoning method. We will see by numerical tests that an optimal solution of (3.9) will not only keep good geometric quality of the grid, but also can track rapid changes in the flow variables.

## 4 Numerical examples

In this section we present different numerical examples to validate the rezoning method of this paper. We begin with the examples where the rezoning is carried out by taking into account the geometric quality of grids only.

### 4.1 Rezoning based on the geometric quality only

#### 4.1.1 Simple domains

First, we check the convergence of our method in the case of simple domains. In Fig. 4, the initial and the rezoned grids generated by the present method are shown, also see [28] on the generation of the initial grids. Obviously, we see that the new rezoning method converges to the uniform grid in both cases. In Figs. 5(a) and (b) the Lagrangian grid generated by Sheshtakov [36] and the rezoned grid obtained by the present method without moving the boundary nodes are given. We find that the cells near the boundary are non-uniform when the boundary nodes are fixed. Fig. 5(c) shows the rezoned grid with movement of the boundary nodes by using the same iterative steps of 5(b), and it is

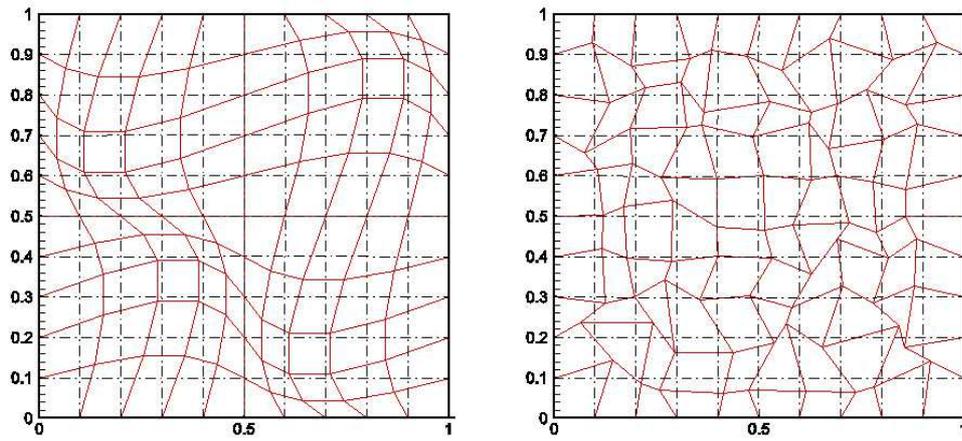


Figure 4: Comparison of Lagrangian and rezoned grids. Left, solid-line: a smooth Lagrangian grid; dash-line: the rezoned grid. Right, solid-line: a random Lagrangian grid with skew cells; dash-line: the rezoned grid.

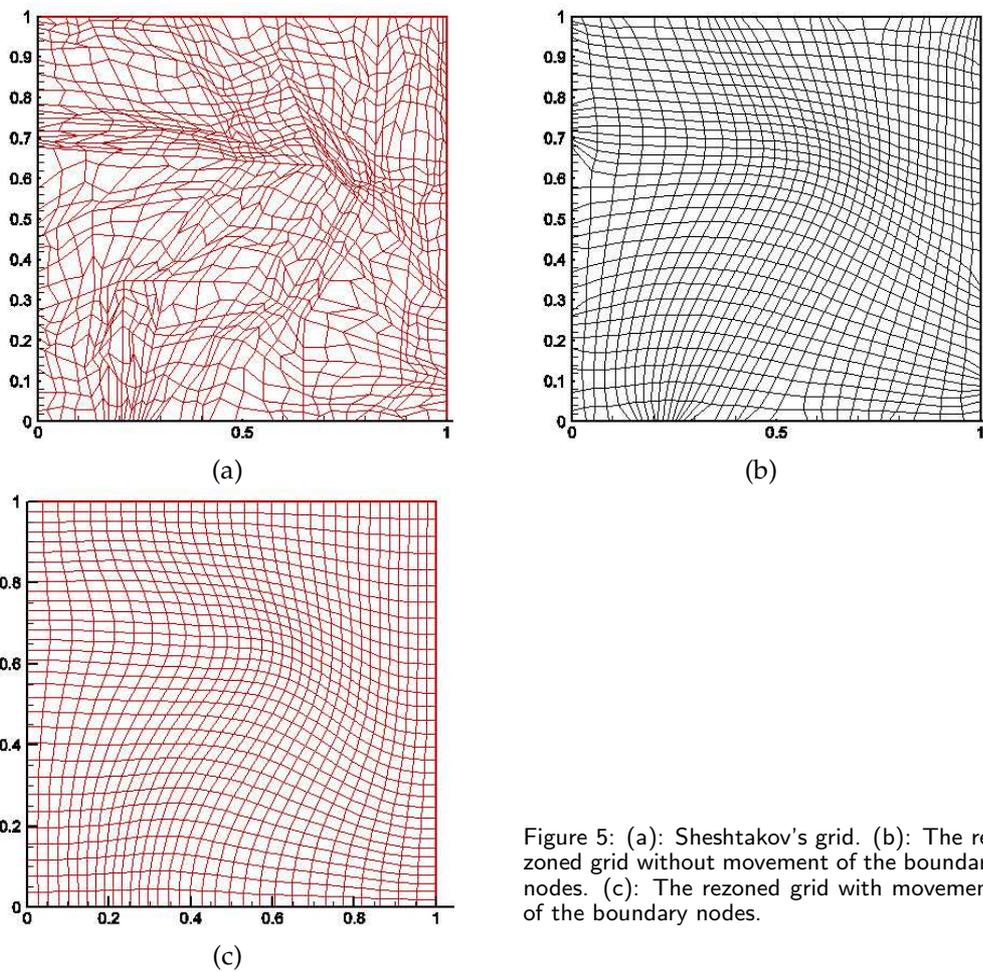


Figure 5: (a): Sheshtakov's grid. (b): The rezoned grid without movement of the boundary nodes. (c): The rezoned grid with movement of the boundary nodes.

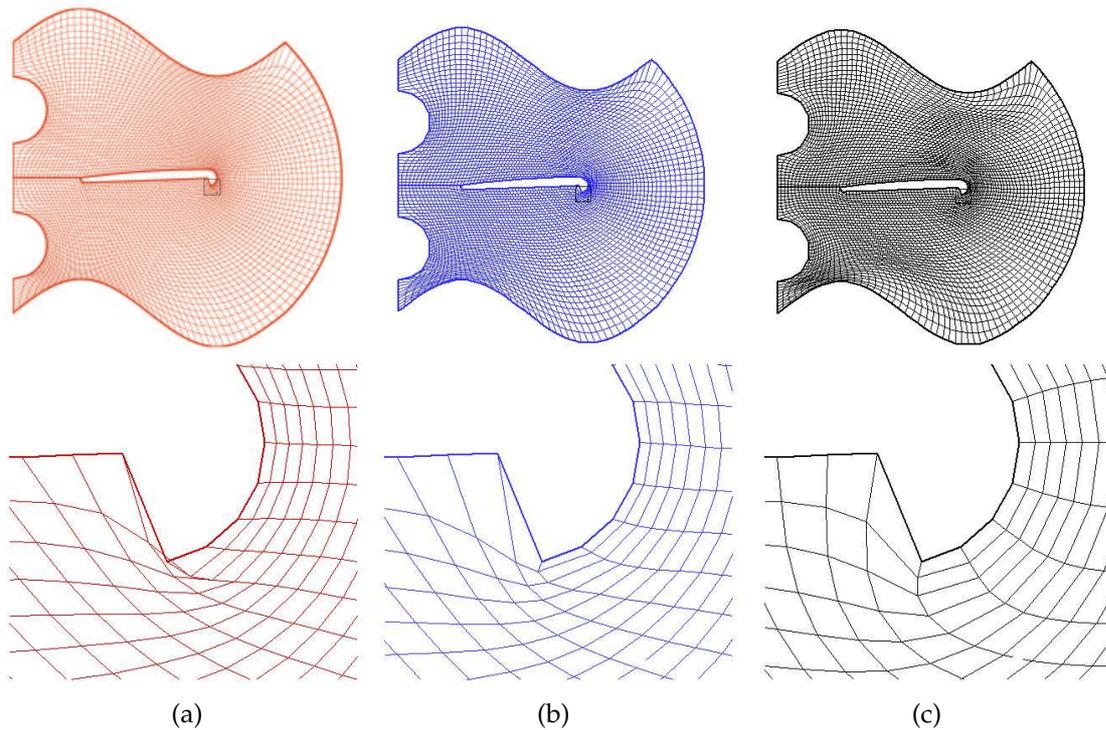


Figure 6: (a) Grid generated using the Winslow method and the close-up of the corner. (b) Grid generated using the GWinslow method and the close-up of the corner. (c) Grid generated using the current method and the close-up of the corner.

clear that the new grid possesses all the good geometric qualities we defined before, and it will converge to the uniform grid if the iterative steps increase. These two examples demonstrate the convergence of the new method.

#### 4.1.2 Complex domains

In this subsection we compare the classical Winslow method [41], the generalized Winslow (GWinslow) method [15] and our rezoning method in complex domains. The first example is shown in Fig. 6 where the corner is a challenge for classical grid generators. In Fig. 6(a), the generated grid using the Winslow method is shown. Obviously, the grid folds at the corner. The grids generated by the GWinslow and our methods are given in Figs. 6(b) and (c), respectively. Clearly, the GWinslow method generates a smooth and convex grid, while the present method gives a grid which is not only convex and smooth, but also uniform and orthogonal.

Numerical results for more complex domains are presented in Figs. 7(a)-(c). From these results we can draw the same conclusion as above. We should point out here that for the harmonic mapping method [30], even though the solution of the harmonic mapping results in an unfolded mapping, the generated grid may fold due to the use of an improper discrete solver, also see [15].

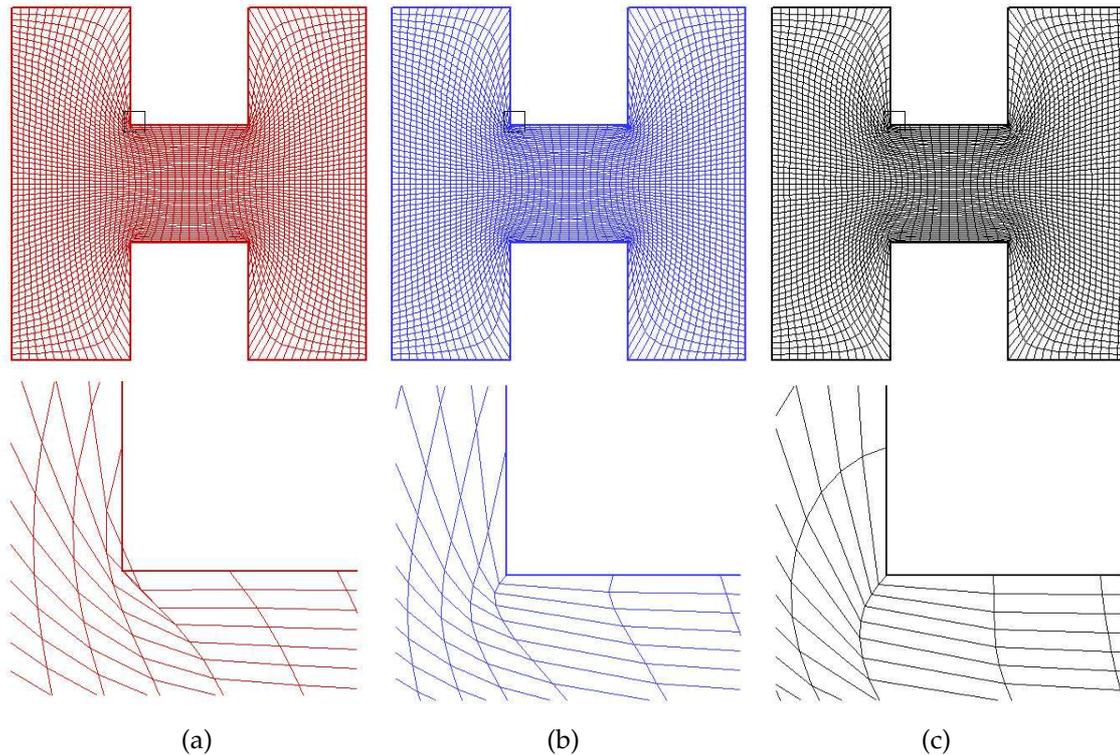


Figure 7: (a) Grid generated by the Winslow method and the close-up of the corner. (b) Grid generated by the GWinslow method and the close-up of the corner. (c) Grid generated by the current method and the close-up of the corner.

## 4.2 Rezoning based on both the geometric quality and adaptation

In this subsection we test our method for two examples in which both the geometric quality and adaptation are taken into account. Using the examples, we will also compare the current method with the mesh redistribution technique used in Tang and Tang [40]. As mentioned in the introduction, the mesh redistribution used in [40] performs well in regular domains. We start with the comparison in simple domains.

In the first example, we generate grids for two given velocity distributions in a unit square (see [40]):

$$u(x,y) = \exp(-8(4x^2 + 9y^2 - 1)^2), \quad (4.1)$$

$$u(x,y) = \exp(-100(y - x^2 + 0.5)^2). \quad (4.2)$$

Generally speaking, the convergence speed may be accelerated or slowed down when increasing or decreasing the parameter  $a$  in the monitor function (3.7). However, if  $a$  is too big, the geometric quality of the generated grid may be affected. An appropriate choice of  $a$  can balance both the convergence speed and the geometric quality of a grid, and the choice of parameters often depends on problems under study and experience.

For the current method, we take  $a=100$  in the monitor function  $f_\rho$ . The numerical results using the mesh redistribution in Tang and Tang [40] and our method are given in Fig. 8. Although it is computationally somewhat more expensive than that used in [40] at each single iterative step, the current method converges faster. As a result, the total computing time for both methods is almost the same. From Fig. 8 we clearly see that the two methods generate very similar grids.

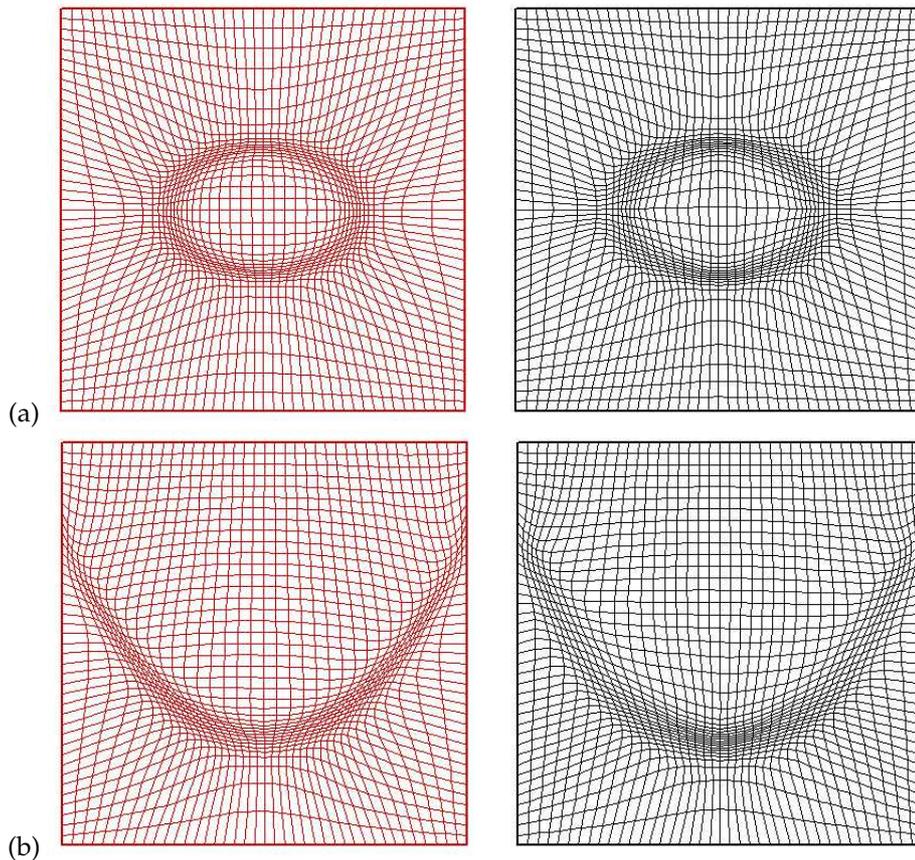


Figure 8: (a) Generated grids for the velocity distribution (4.1). Left: the mesh redistribution given by [40]. Right: The current method. (b) Generated grids for the velocity distribution (4.2). Left: The mesh redistribution given by [40]. Right: The current method.

In the second example, we compare both methods in a complex concave domain shown in Fig. 9. We give a velocity distribution function

$$u(x,y) = \exp(-8(9(x-3)^2 + 9(y-3)^2 - 1)^2) \quad (4.3)$$

in this complex domain. The numerical results using both methods are presented in Figs. 9(a) and (b), respectively. From Fig. 9(a) we find that the grid generated by the mesh redistribution in [40] is folded at the four corners, while Fig. 9(b) clearly shows that the

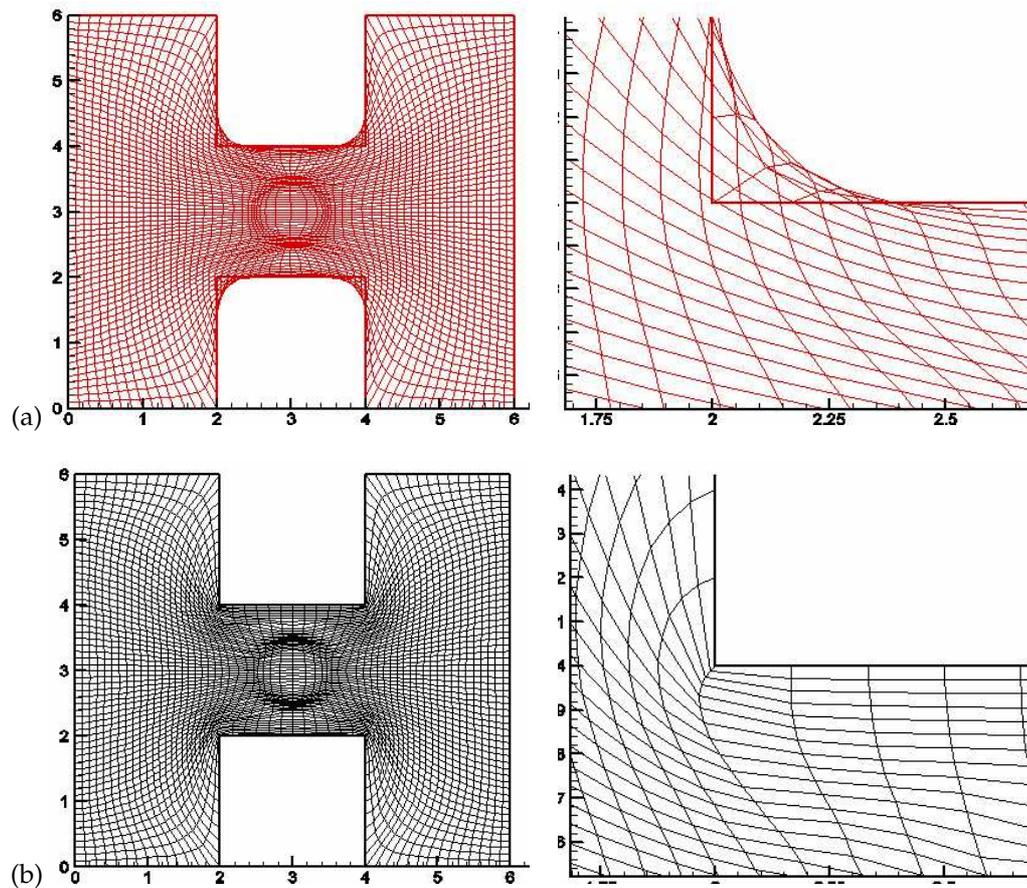


Figure 9: (a) Grid generated by the mesh redistribution used in [40] and the close-up of the corner. (b) Grid obtained using the current method and the close-up of the corner.

grid generated using the current method not only well tracks the variation of the velocity distribution, but also keeps good geometric quality.

### 4.3 Application to an ALE method

In this subsection we incorporate our rezoning method in the ALE method described in Section 2 to numerically solve three well-known fluid Benchmark problems. We will compare our ALE method with the Lagrangian method to validate robustness and adaptation of the current rezoning method. For all the numerical examples below, the rezoning phase in the ALE computation is carried out once every ten time steps, and the parameters  $a$  and  $b$  in the monitor function  $f_\rho$  are taken to be 100.

#### A. SEDOV'S BLAST WAVE PROBLEM [1]

Sedov's blast wave problem models the blast wave from an intense explosion in a

perfect gas, so is an example of diverging shock waves. Consider a uniform medium in space with zero initial pressure. A fixed amount of energy is deposited initially at the origin. As time increases, a blast wave expands away from the origin. Because the initial pressure is zero, the shock associated with the blast wave is infinite in strength, and a similarity solution for the post shock profile can be obtained. The solution was first found by Sedov in 1959, and is particularly useful for testing the accuracy of multidimensional computational schemes [1], and has been a standard Benchmark problem for Lagrangian methods. Here we run a two-dimensional calculation using our ALE code with the current rezoning method. An illustrative choice for a computational mesh is the one in which the plane coordinates are  $\Delta x = \Delta y = 1/45$  in a quadrant. A single unit of energy is deposited in the central cell of the mesh. According to the analytic theory, the blast shock wave should expand with radius equal to 1 at time  $t = 1$  and the maximum of the density is 6.0.

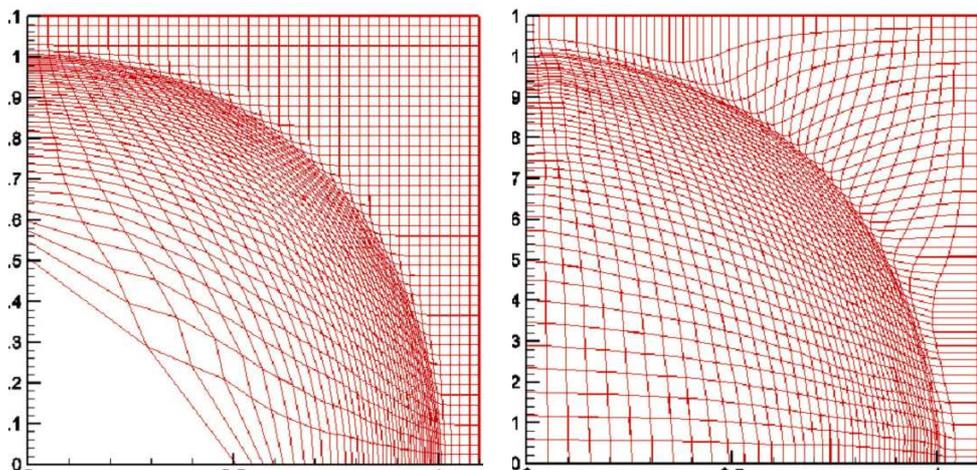


Figure 10: Grids at  $t = 1.0$  for Sedov's problem. Left: Lagrangian method. Right: ALE method.

The computational grids and the density profile at  $t = 1.0$  obtained by using the ALE and Lagrangian methods are shown in Figs. 10 and 11, respectively. Comparing with the analytic solutions, it is apparent that both methods perform well in capturing the location of the shock wave. However, the ALE method is more accurate than the Lagrangian one in the vicinity of the maximal density. Moreover, the Lagrangian grid becomes skew, while the ALE grid not only keeps good geometric quality but also tracks the shock wave as the Lagrangian method does. Fig. 12 shows the grids of the ALE and Lagrangian methods at  $t = 2$ . From Fig. 12 we see that the same conclusion as at  $t = 1$  can be drawn.

#### B. SALTZMAN'S PISTON PROBLEM [35]

This problem tests the ability of numerical methods to model shock waves that are oblique to the mesh. The set-up of the problem is the following. An ideal gas with the specific heat ratio  $\gamma = 5/3$  is filled in the unit box, one end of which is a movable piston. The piston moves into the box with a constant velocity of 1.0 and a strong shock

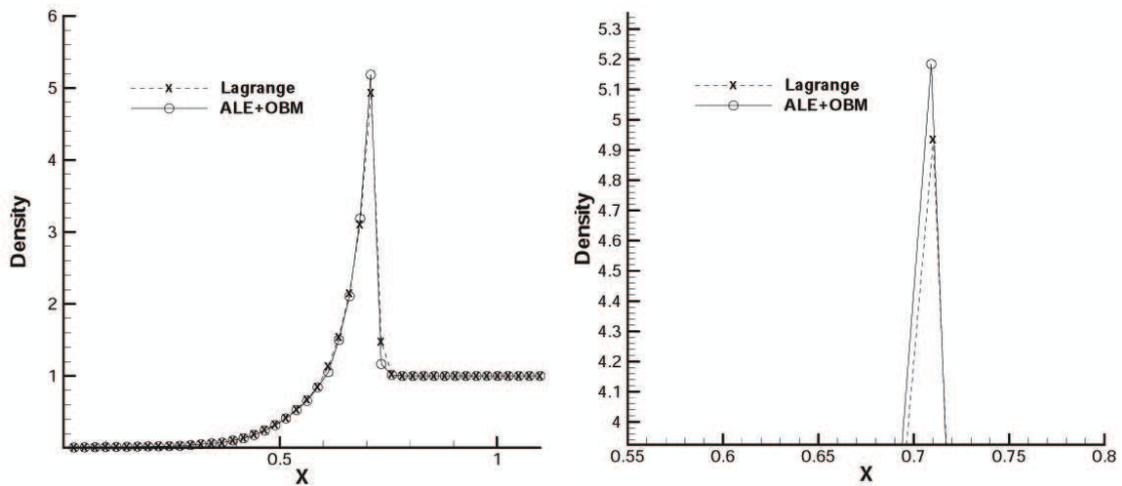


Figure 11: Density profile at  $t=1.0$  for Sedov's problem. Left: whole domain. Right: close-up near the maximal value.

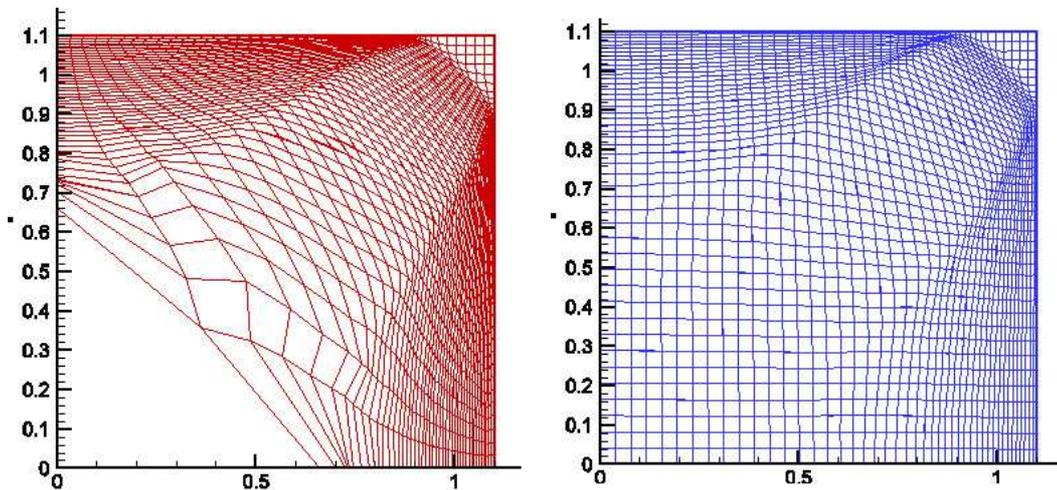


Figure 12: Grids at  $t=2.0$  for Sedov's problem. Left: Lagrangian method. Right: ALE method.

wave is generated from the moving end. On the upper and lower boundaries the symmetric boundary conditions are used. Initially, we use a  $100 \times 10$  grid with the nodal  $x$  coordinates

$$x_{ij} = (i-1)\Delta x + (11-j) \sin\left(\frac{\pi(i-1)}{100}\right)\Delta y$$

and the uniform nodal  $y$  coordinates.

In Figs. 13 and 15 the grids of the Lagrangian and ALE methods at  $t=0.5$  and  $t=0.8$  are presented, respectively. It is obvious to observe that the Lagrangian grid is getting more and more skew, while the ALE grid preserves the good geometric quality. Fig. 14

shows the pressure profiles of both methods at  $t=0.5$ . We find that the computed shock wave by the Lagrangian method is smeared due to skewness of the grid, while the ALE method resolves the shock wave well.

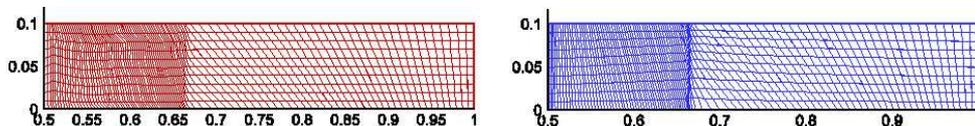


Figure 13: Grids at  $t=0.5$  for Saltzman's problem. Left: Lagrangian method. Right: ALE method.

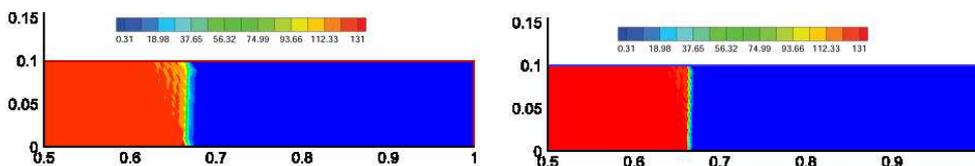


Figure 14: Pressure profiles at  $t=0.5$  for Saltzman's problem. Left: Lagrangian method. Right: ALE method.

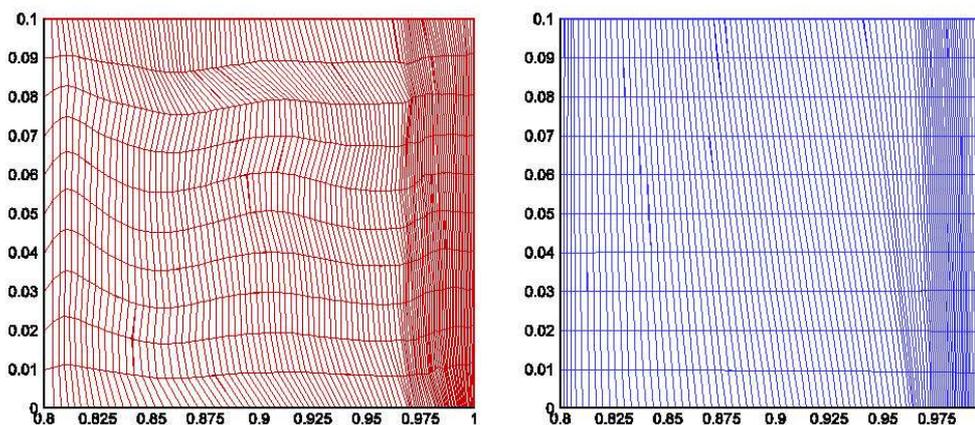


Figure 15: Grids at  $t=0.8$  for Saltzman's problem. Left: Lagrangian method. Right: ALE method.

### C. DUKOWICZ'S PROBLEM [18]

This problem is a well-known shock refraction problem which tests the ability of numerical methods for multi-fluid simulations. The initial mesh is composed of two adjacent regions, each initially containing ideal gases with  $\gamma = 1.4$  of different densities but equal pressures, see Fig. 16. The nominal conditions are: incident shock with Mach number  $M_s = 2$ , interface density ratio  $\rho_2/\rho_1 = 1.5$ , and shock-interface angle of incident  $= 60^\circ$ . Region 1 has a  $36 \times 30$  mesh, with the left boundary vertical and the right boundary slanted at  $60^\circ$  to represent the interface. Region 2 has a  $40 \times 30$  mesh uniformly slanted at  $60^\circ$ . The upper and lower boundaries are reflective, and the left boundary is a piston, which moves to the right with a velocity of 1.48 units, driving a Mach 2 shock into Region

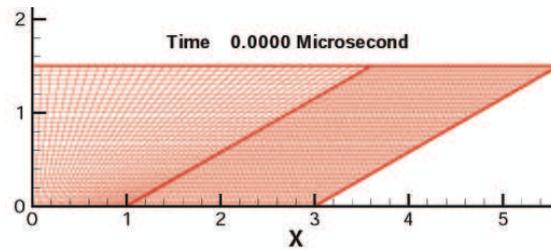


Figure 16: Grids at  $t=0.0$  for Dukowicz's problem.

1. Fig. 17 shows that the density profile of the Lagrangian and ALE methods at  $t = 1.1$ . We find that both methods can track the material interface quite intactly. However, the computed shock wave by the Lagrangian method is somewhat smeared due to skewness of the grid, while the ALE method well resolves the shock wave. Figs. 18 and 19 show the grids of two methods at  $t = 1.1$  and the close-up of the material interface, respectively. It is obvious that the Lagrangian grid becomes skew, while the ALE grid not only preserves good geometrical quality but also moves the cells towards the shock wave.

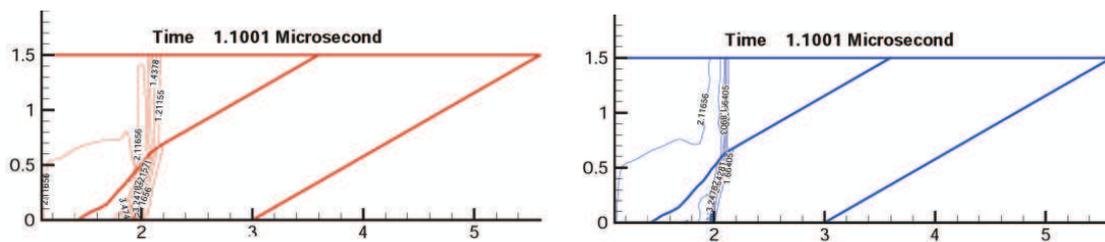


Figure 17: Density contour at  $t = 1.1$  for Dukowicz's problem. Left: Lagrangian method. Right: ALE method.

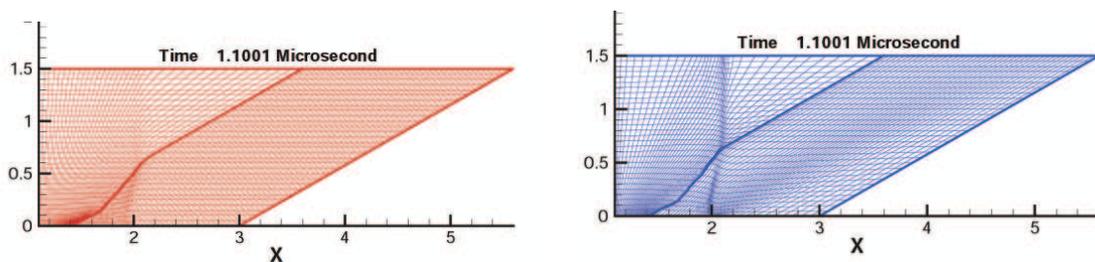


Figure 18: Grids at  $t = 1.1$  for Dukowicz's problem. Left: Lagrangian method. Right: ALE method.

## 5 Conclusions

Based on the theory of optimization, we use edges and angles of cells to represent the geometric quality of a computational grid, and utilize the local gradients of the flow vari-

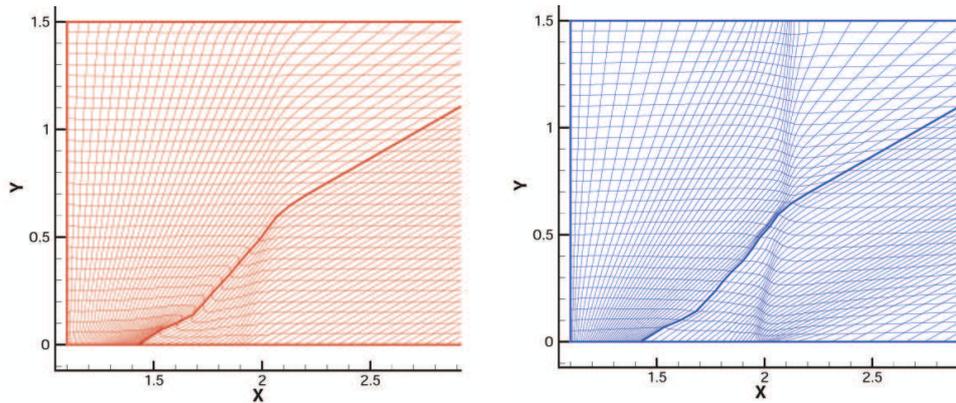


Figure 19: Grids for Close-up near the interface at  $t = 1.1$  for Dukowicz's problem. Left: Lagrangian method. Right: ALE method.

ables to describe the variation of flow field to construct a multi-objective programming model. The solution of this optimization problem controls movement of the grid in accordance to suitable balance between the geometric quality and adaptation of the grid. By solving this optimization problem, we have thus obtained a new grid rezoning (grid movement) method, which not only keeps good geometric quality of the grid, but also tracks rapid changes in the flow field. At the same time, we have also incorporated the rezoning method of this paper into an ALE method which is widely used in the simulation of high-speed multi-material flows, and obtained a new ALE method. A number of numerical examples, including some complex concave domains and multi-material flows, have been carried out. The comparison of the current rezoning method with a few others has been given. The numerical results demonstrate the robustness and adaptation of the current rezoning and ALE methods.

## Acknowledgments

This work is supported by the Special Funds for Major State Basic Research Projects (Grant No. 2005CB321700), CAEP (Grant No. 20060108) and NSFC of China.

## Appendix: Proof of Propositions 3.1 and 3.2

**PROOF OF PROPOSITION 3.1.** Given an octagon  $ABCDEFGH$  and a point  $P$  in it. Then four quadrilaterals  $ABPH$ ,  $BCDP$ ,  $DEFP$  and  $FGHP$  are obtained (see Fig. 20). For an angle we define its value in anticlockwise, for instance in Fig. 21, the angle  $\angle ROZ$  has the beginning edge  $OR$  and ending edge  $OZ$ , and one has  $\angle ROZ \in [0, 2\pi)$  and  $\angle ZOR = 2\pi - \angle ROZ$ . The angles affected by movement of  $P$  are  $\angle ABP$ ,  $\angle PBC$ ,  $\angle CDP$ ,  $\angle PDE$ , and  $\angle EFP$ ,  $\angle PFG$ ,  $\angle GHP$ ,  $\angle PHA$ ,  $\angle BPH$ , and  $\angle HPF$ ,  $\angle FPD$ ,  $\angle DPB$ .

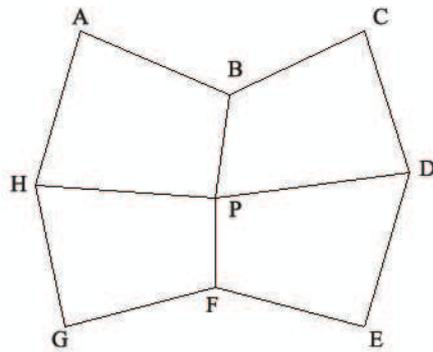


Figure 20: Octagon  $ABCDEFGH$ , four quadrilaterals meeting at a point  $P$ .

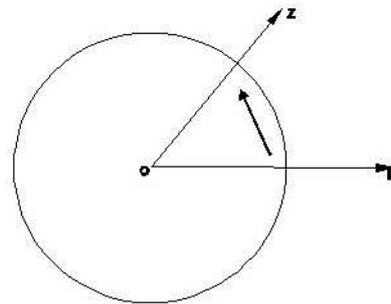


Figure 21: Angle  $\angle ROZ$  with the beginning edge  $OR$  and the ending edge  $OZ$ .

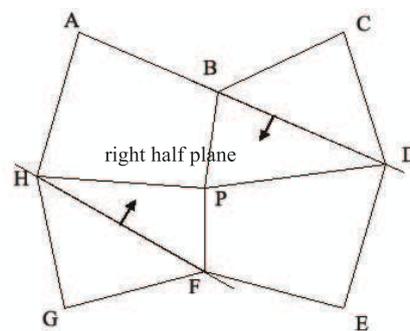
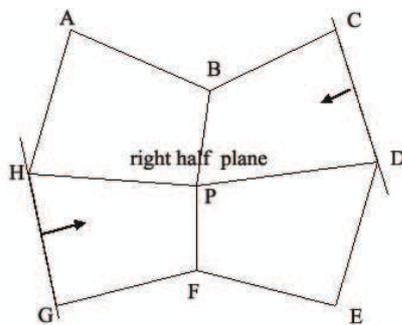


Figure 22: Right half planes defined by the segments  $HG$  and  $HF$ .

The eight sides of the octagon and the four diagonals  $HB, BD, DF, FH$  form a set of 12 segments. Thus, any straight line which covers a segment divides the plane  $\mathbb{R}^2$  into two half planes. We define the half plane as the right half plane the inner normal vector of which directs to the inside of the octagon as shown in Fig. 22. Totally, we have such 12 right half planes. We denote by  $\mathcal{S}^*$  the intersection of these 12 right half planes. Since any half plane is convex and the intersection of convex sets is still convex, the set  $\mathcal{S}^*$  is convex. Next, we prove that  $\mathcal{S}^*$  is in fact the feasible set  $\mathcal{S}$  (if there exists a feasible set  $\mathcal{S}$ ).

(i) We first show that if  $P \in \mathcal{S}^*$ , then  $P \in \mathcal{S}$ . To prove this, it is easy to see that it suffices to show that the angles  $\angle ABP, \angle PBC, \angle CDP, \angle PDE, \angle EFP$ , and  $\angle PFG, \angle GHP, \angle PHA, \angle BPH, \angle HPF, \angle FPD$ , and  $\angle DPB$  are all less than  $\pi$ . To this end, we divide these 12 angles into two types: Type I consists of the angles with the vertex  $P$ , while the rest of the 12 angles constitutes Type II.

In the case of Type I, we only prove  $\angle HPF < \pi$ , and the same conclusion for the other angles in Type I can be obtained in the same manner. If  $P$  locates at the right half plane of the straight line containing the line  $FH$ , then  $\angle HPF$  is an interior angle of the triangle  $\Delta FPH$ , and therefore  $\angle HPF$  is less than  $\pi$ . Otherwise,  $\angle FPH$  is an interior angle of the triangle  $\Delta FPH$ , and hence,  $\angle HPF = 2\pi - \angle FPH$  is larger than  $\pi$ .

In the case of Type II, we only show  $\angle EFP < \pi$ , and the same conclusion for the rest angles in Type II can be proved in the same way. If  $P$  locates at the right half plane of the straight line containing the line  $EF$ ,  $\angle EFP$  is an interior angle of the triangle  $\Delta EFP$ , and therefore  $\angle EFP$  is less than  $\pi$ . Otherwise,  $\angle PFE$  is an interior angle of the triangle  $\Delta EFP$ , and hence  $\angle EFP$  is larger than  $\pi$ .

Therefore, we have proved  $\mathcal{S}^* \subset \mathcal{S}$ .

(ii) Let  $P \in \mathcal{S}$ . From (i) we can easily see that the proof procedure in (i) is reversible. Hence, we have  $P \in \mathcal{S}^*$ , i.e.,  $\mathcal{S} \subset \mathcal{S}^*$ .

Thus,  $\mathcal{S} = \mathcal{S}^*$ , and the proof of Proposition 3.1 is complete.  $\square$

**PROOF OF PROPOSITION 3.2.** From the definition we can verify that  $f_\alpha^P, f_l^P > 0$ . Hence,  $f_\alpha^P \cdot f_l^P > 0$  and  $f_\alpha^{P^*} \cdot f_l^{P^*} > 0$  has the infimum. Thus, there exists an optimal solution of the problem (3.2).

On the other hand, assume that an optimal solution  $P \in \mathcal{S}$  to the problem (3.2) is not a non-inferior solution to the problem (3.1). Then, by virtue of the definition of non-inferior solutions, there exists a point  $P^* \in \mathcal{S}$ , such that  $f_\alpha^{P^*} \leq f_\alpha^P$  and  $f_l^{P^*} \leq f_l^P$ , and at least one of two inequalities holds strictly. Therefore,  $f_\alpha^{P^*} \cdot f_l^{P^*} < f_\alpha^P \cdot f_l^P$ . This obviously contradicts the assumption of the optimality of  $P$ . The proof of Proposition 3.2 is complete.  $\square$

## References

- [1] F.L. Addressio, R. Baumgardner, J.K. Dukowicz, N.L. Johnson, B.A. Kashiwa, R.M. Rauenzahn and C. Zemach, CAVEAT: A computer code for fluid dynamics problems with large distortion and internal slip. LANL Report, LA-10613-MS, 1992.
- [2] I. Altas and J.W. Stephenson, A two-dimensional adaptive mesh generation method. J. Comput. Phys. 94 (1991), 201-224.
- [3] A.A. Amsden and C.W. Hirt, A simple scheme for generating general curvilinear grids. J. Comput. Phys. 11 (1973), 348-359.
- [4] R.W. Anderson, N.S. Elliott and R.B. Pember, An arbitrary Lagrangian-Eulerian method with adaptive mesh refinement for the solution of the Euler equations. J. Comput. Phys. 199 (2004), 598-617.
- [5] B.N. Azarenok, Variational barrier method of adaptive grid generation in hyperbolic problems of gas dynamics. SIAM J. Numer. Anal. 40 (2002), 651-682.
- [6] B.N. Azarenok, S.A. Ivanenko and T. Tang, Adaptive mesh redistribution method based on Godunov's scheme. Comm. Math. Sci. 1 (2003), 152-179.
- [7] B.N. Azarenok and T. Tang, Second-order Godunov-type scheme for reactive flow calculations on moving meshes, J. Comput. Phys. 206 (2005), 48-80.
- [8] D.J. Benson, An efficient, accurate, simple ALE method for nonlinear finite element programs. Computer Meth. Appl. Mech. Eng. 72 (1989), 305-350.
- [9] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes. Computer Meth. Appl. Mech. Eng. 99 (1992), 235-394.
- [10] J.U. Bracknill, An adaptive grid with directional control. J. Comput. Phys. 108 (1993), 38-50.
- [11] J.U. Bracknill and J.S. Saltzman, Adaptive zoning for singular problems in two dimensions. J. Comput. Phys. 46 (1982), 342-368.

- [12] J.U. Brackbill, Coordinate system control: Adaptive meshes. In: Numerical Grid Generation, ed. by J.F. Thompson, North-Holland, New York, 1982, pp. 277-294.
- [13] 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), 221-244.
- [14] 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), 1978-1994.
- [15] A.A. Charakhch'yan and S.A. Ivanenko, A variational form of the Winslow grid generator. J. Comput. Phys. 136 (1997), 385-398.
- [16] J.B. Cheng, G.P. Zhao, Z.P. Jia, Y.B. Chen, J.X. Cheng, S.H. Wang and W.Z. Wen, Sliding free Lagrangian-Eulerian finite element method. In: Computational Science – ICCS 2006, Lecture Notes in Computer Science, Vol. 3991, Springer Verlag, Berlin, Heidelberg, 2006, pp. 851-855.
- [17] G.S. Collins and H.J. Melosh, <http://www.lpl.arizona.edu/tektion/sales2.html>. SALES2: A multi-material extension to the SALE hydrocode with improved equation of state and constitutive model.
- [18] J.K. Dukowicz and B.Z.A. Meltz, Vorticity errors in multidimensional Lagrangian codes. J. Comput. Phys. 99 (1992), 115-134.
- [19] M.S. Gadala, Recent trends in ALE formulation and its applications in solid mechanics. Computer Meth. Appl. Mech. Eng. 193 (2004), 4247-4275.
- [20] J.O. Hallquist, User's manual for DYNA2D – An explicit two-dimensional hydrodynamic finite element code with interactive rezoning and graphical display. UCID-18756 Rev. 1, 1982.
- [21] W.H. Hui, The unified coordinate system in computational fluid dynamics. Commun. Comput. Phys. 2 (2007), 577-610.
- [22] X.D. Hang, Parallel Iterative Solution of Partial Differential Equations and Mesh Optimization. Ph.D. Thesis, Graduate School of CAEP, Beijing, 2004.
- [23] M.J. van Haaren, H.C. Stoker, A.H. van den Boogaard and J. Huétink, The ALE-method with triangular elements: Direct convection of integration point values. Inter. J. Numer. Meth. Eng. 49 (2000), 697-720.
- [24] D.F. Hawken, J.J. Gottlieb and J.S. Hansen, Review of some adaptive node-movement techniques in finite-element and finite difference solutions of partial differential equations. J. Comput. Phys. 95 (1991), 254-302.
- [25] 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), 227-253.
- [26] W. Huang, Y. Ren and R.D. Russell, Moving mesh partial differential equations (MMPDEs) based on the equidistribution principle. SIAM J. Numer. Anal. 31 (1994), 709-730.
- [27] C.Q. Jin and K. Xu, Numerical study of the unsteady aerodynamics of freely falling plates, Commun. Comput. Phys. 3 (2008), 834-851.
- [28] P. Knupp, L.G. Margolin and M. Shashkov, Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods. J. Comput. Phys. 176 (2002), 93-128.
- [29] P. Knupp and S. Steinberg, Fundamentals of Grid Generation. CRC Press, Boca Raton, 1993.
- [30] R. Li, T. Tang and P.W. Zhang, Moving mesh methods in multiple dimensions based on harmonic maps. J. Comput. Phys. 170 (2001), 562-588.
- [31] K. Lipnikov and M. Shashkov, The error-minimization-based strategy for moving mesh methods. Commun. Comput. Phys. 1 (2006), 53-81.
- [32] I. Lomtev, R.M. Kirby and G.E. Karniadakis, A discontinuous Galerkin ALE method for compressible viscous flows in moving domains. J. Comput. Phys. 155 (1999), 128-159.
- [33] L.G. Margolin, Introduction to an arbitrary Lagrangian-Eulerian computing method for all flow speeds. J. Comput. Phys. 135 (1997), 198-202.

- [34] J.S. Peery and D.E. Carroll, Multi-Material ALE methods in unstructured grids. *Computer Meth. Appl. Mech. Eng.* 187 (2000), 591-619.
- [35] J. Saltzman and P. Collella, Second order upwind transport methods for Lagrangian hydrodynamics. LANL Report, LA-UR-85-678, 1985.
- [36] A.I. Shestakov, J.A. Harte and D.S. Kershaw, Solution of the diffusion equation by the finite elements in Lagrangian hydrodynamic codes. *J. Comput. Phys.* 76 (1998), 385-413.
- [37] R.W. Smith and J.A. Wright, An implicit edge-based ALE method for the incompressible Navier-Stokes equations. *Int. J. Numer. Meth. Fluids* 43 (2003), 253-279.
- [38] J.L. Steger and D.S. Chausee, Generation of body-fitted coordinates using hyperbolic partial differential equations. *SIAM J. Sci. Stat. Comput* 1 (1980), 431-437.
- [39] H.Z. Tang, A moving mesh method for the Euler flow calculations using a directional monitor function. *Commun. Comput. Phys.* 1 (2006), 656-676.
- [40] 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.
- [41] A. Winslow, Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh. *J. Comput. Phys.* 1 (1967), 149-172.
- [42] Y.Z. Yao, Research on Grid Optimization for Computational Fluid Dynamics. Ph.D. Thesis, Graduate School of CAEP, Beijing, 2007.