Journal of Computational Mathematics
Vol.31, No.6, 2013, 592–619.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1307-m4182

# A REMAPPING METHOD BASED ON MULTI-POINT FLUX CORNER TRANSPORT UPWIND ADVECTION ALGORITHM*

Zhijun Shen
*National Key Laboratory of Science and Technology on Computational Physics, Institute of Applied Physics and Computational Mathematics, P.O.Box 8009-26, Beijing 100088, China*
*and Center for Applied Physics and Technology, HEDPS, Peking University, Beijing 100871, China*
*Email: shen_zhijun@iapcm.ac.cn*
Guixia Lv
*National Key Laboratory of Science and Technology on Computational Physics, Institute of Applied Physics and Computational Mathematics, P.O.Box 8009-26, Beijing 100088, China*
*Email: lvguixia@126.com*

**Abstract**

A local remapping algorithm for scalar function on quadrilateral meshes is described. The remapper from a distorted grid to a rezoned grid is usually regarded as a conservative interpolation problem. The present paper introduces a pseudo time to transform the interpolation into an initial value problem on a moving grid, and construct a moving mesh method to solve it. The new feature of the algorithm is the introduction of multi-point information on each edge, which leads to the numerical flux consistent with grid node motion. During the procedure of deriving scheme, we illustrate a framework about how the algorithms on a rectangular mesh are easily generated to those on a moving mesh. The basic ideas include: (i) introducing coordinate transformation, which maps the irregular domain in physical space to a perfectly regular computational domain, and (ii) deriving finite volume methods in the physical domain, which can be viewed as a discretization of the transformed equation. The resulting scheme is second-order accurate, conservative and monotonicity preserving. Numerical examples are carried out to show the good performance of our schemes.

*Mathematics subject classification:* 65D05, 76M12, 34M25.
*Key words:* Remapping, Advection, Multi-point flux, Coordinate transformation, Geometric conservation law.

## 1. Introduction

In numerical simulations of fluid flow, the arbitrary Lagrangian Eulerian method (ALE) has been regarded as having excellent accuracy, robustness, or computational efficiency compared with Euler and Lagrangian method. It is usual to be separated into three phases. These are: (1) a Lagrangian phase in which the solution and grid are updated; (2) a rezoning phase in which the nodes of the computational grid (old) are moved to more optimal positions (new); and (3) a remapping phase in which the solution is mapped from a distorted Lagrangian gird onto the rezoned grid. Hence the remapping algorithm is a very important part in ALE method.

Given a distorted Lagrangian grid and a rezoned grid, there are two kinds of classical remapping methods. One is finding the intersections of each new cell with the old ones. Such method

* Received May 26, 2012 / Revised version received January 15, 2013 / Accepted July 9, 2013 / Published online October 18, 2013 /

is suitable for the problems in which the two grids are independent even have different topologies. Recently such methods have been extensively used in multi-material flows calculation with interface reconstruction or reconnection-based ALE methods [11] [12] [20]. Finding the intersections is feasible but computationally very expensive in two dimensions, and not practical in three dimensions due to its complication. Another remapper usually termed as continuous remapping is constructed by advection algorithms, which can avoid detailed calculations of the intersections. The underlying assumptions are that the topology of the mesh is fixed and the mesh motion during a step is less than the dimensions of the elements.

There is an extensive literature about advection algorithms, cf. [9] [40] [29] [1] [2] [5] [30] [26]. The most widely used is the donor cell upwind (DCU) method in which the advected quantity only streams from the adjacent cell on the upwind side (the donor cell). For a structured quadrilateral grid, it has five-point stencils. Such method is accurate and robust in most situations, but sometimes it may suffer from some small flaws. For example, suppose the flow is two-dimensional, and some physical quantities should be transported between grid cells sharing only a vertex. If only the one-dimensional advection algorithms are applied simultaneously in the two mesh directions, the velocity at which a signal propagates for advection along the diagonal may be slower than in the two mesh directions [19]. To alleviate or cure this kind of error, the corner transport upwind (CTU) method proposed by Colella [5] is a good choice, which is based on tracing the characteristics of the advection equation in two dimensions. The CTU scheme involves more information, such as nine-point stencils in structured quadrilateral grid, hence has larger Courant number compared with the DCU method. The same algorithm was derived in a different manner by van Leer [16]. Dukowicz and Baumgardner put forward a kind of new method with corner contributions [9].

In remapping algorithm framework based on advection approach, a local remapper exchanging conservation quantities between neighboring cells is extensively used [12] [3] [15] [24] [25] [28] [31] [33] [39]. Among them, Pember and Anderson [33] proposed a corner transport method. Since the remapping algorithm presented in [33] is only a middle procedure in solving ALE problems, some details and numerical results of this algorithm are omitted. In addition, all of the above schemes do not consider grid node moving information and use single-edge flux at cell interface, which may result in large errors in some situations (see case 1 in numerical experiments, below). P. Hoch et.al. [14] considered such case, and computed two sub-volumes of fluxing for an edge for both adjacent cell. But the method has not to be generalized to the higher-order accurate case and the details are neglected.

In this paper, we hope to benefit from all previous experiences and develop a second-order accurate CTU method for solving remapping problem. Analogously with [31] [33], we introduce a pseudo time and transform the interpolation into an initial value problem (hereafter we call it remapping equation) on a moving grid. However, we adopt a wave propagation method [18] which may be easier than that in [33] to extend to solving more complicated system of nonlinear conservation laws in moving grid context, especially for those nonlinear equations in non-conservative form, such as Elasticity equations or multi-phase fluid problems. The main new feature of our remapping scheme compared with traditional ones is to introduce 'node velocity' and two half edge fluxes per cell interface which is consistent with node motion manner. Such technique has been extensively used in solving Lagrangian form hydrodynamics, cf [7] [23] [21] [22] [4], but does not appear in other moving mesh method context. Different from [14], we need not to compute self-tangled patch created by edge displacement. At the same time, a high order CTU method is implemented.

The 'node velocity' in remapping problem results from the introduction of the pseudo time and avoids the difficulty of evaluating exact intersection areas of two layer grid. We design an elaborate example to show the drawbacks of traditional single-edge flux method and advantages of our algorithm.

Although this paper aims to design remapping algorithms, the remapping problem is not introduced immediately at the beginning of the paper for the sake of clarity. An outline of this paper is as follows. We recall two kinds of classical advection methods on a fixed grid in Section 2, including the DCU, CTU algorithms and their high-order accurate correction on quadrilateral grid. In Section 3, we make a modification to the classical DCU and CTU method when discretizing advection equation, in which multi-point flux at edge of a grid cell is introduced. In Section 4, we discuss the remapping methods. Firstly, the physical pictures of an advection equation and the remapping equation are described. Then a finite volume method with the first order accurate DCU flux for solving the remapping equation is recalled, and a coordinate transformation approach equivalent to the DCU scheme is depicted. Also the CTU and its second-order accurate modification schemes are derived. In the end, a multi-point flux remapping scheme according to the modification in Section 3 is constructed. In Section 5, some two-dimensional remapping simulations on different sequences of grids are performed. The results are shown graphically and analyzed from the viewpoints of accuracy and order of convergence. Some concluding remarks are given in Section 6.

## 2. The Advection Algorithm

Consider a two-dimensional variable-coefficient advection equation

$$q_t + (uq)_x + (vq)_y = 0, \tag{2.1}$$

where $q(x, y, t)$ is a conservative quantity, and $u(x, y), v(x, y)$ are advection velocities along $x$ and $y$ direction, respectively.

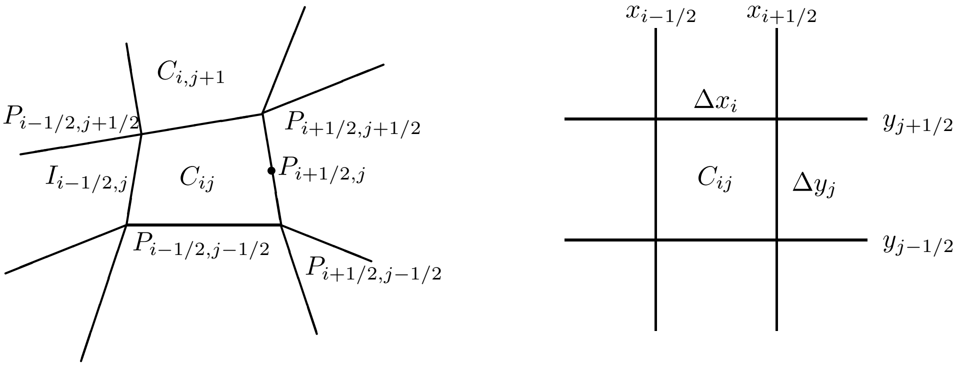

Fig. 2.1. Notations on (a) a quadrilateral grid and (b) a rectangular grid.

The finite volume approach can be applied on any shape control volume $C$. In this paper, we assume that the velocities $\vec{u} = (u, v)$ are defined at the edge of grid and limit our method on logically rectangular grids. We adopt the following notations. A grid cell $C_{i,j}$ has vertices (points) $P_{i-1/2,j-1/2}$, $P_{i+1/2,j-1/2}$, $P_{i+1/2,j+1/2}$ and $P_{i-1/2,j+1/2}$. The coordinates of a vertex $P_{i+1/2,j+1/2}$ are $(x_{i+1/2,j+1/2}, y_{i+1/2,j+1/2})$. The edge connecting points $P_{i+1/2,j-1/2}$ and $P_{i+1/2,j+1/2}$ is denoted by $I_{i+1/2,j}$, its midpoint is $P_{i+1/2,j}$, and the flux defined at the edge is $F_{i+1/2,j}$. Similarly, the edge connecting points $P_{i-1/2,j+1/2}$ and $P_{i+1/2,j+1/2}$ is $I_{i,j+1/2}$, and the flux is $G_{i,j+1/2}$, see Fig. 2.1(a). A special case is a rectangular grid cell of the form $C_{i,j} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$, where $x_{i+1/2} - x_{i-1/2} = \Delta x_i$ and $y_{j+1/2} - y_{j-1/2} = \Delta y_j$, see Fig. 2.1(b).

### 2.1. Finite volume approach

Using the integral form of the conservation law on grid $C_{i,j}$ in Fig. 2.1(a), there is

$$Q_{ij}^{n+1} = Q_{ij}^n - \frac{1}{|C_{i,j}|}\int_{t^n}^{t^{n+1}}\int_{\partial C_{i,j}} (u,v)\cdot\vec{n}(s) q(x(s),y(s),t)dsdt,$$

where $Q_{ij}^n$ represents the cell average of quantity $q$ over this cell at time $t^n$, $|C_{i,j}|$ is the area of the cell $C_{i,j}$, and $\vec{n}(s)$ is the outward vector normal to the edge $(x(s), y(s))$ of $C_{i,j}$ parameterized by the arclength $s$.

The fully discrete finite volume method in flux form is

$$\begin{aligned} Q_{ij}^{n+1} = Q_{ij}^n - \frac{\Delta t}{|C_{i,j}|}(&F_{i+1/2,j}|I_{i+1/2,j}| - F_{i-1/2,j}|I_{i-1/2,j}| \\ &+G_{i,j+1/2}|I_{i,j+1/2}| - G_{i,j-1/2}|I_{i,j-1/2}|), \end{aligned} \tag{2.2}$$

where $|I_{i+1/2,j}|$ is the length of the interface $I_{i+1/2,j}$, and the fluxes are

$$F_{i+\frac{1}{2},j} = \frac{1}{\Delta t|I_{i+1/2,j}|}\int_{t^n}^{t^{n+1}}\int_{I_{i+1/2,j}} (u,v)\cdot\vec{n}(s)q(x(s),y(s),t)dsdt, \tag{2.3a}$$

$$G_{i,j+\frac{1}{2}} = \frac{1}{\Delta t|I_{i,j+1/2}|}\int_{t^n}^{t^{n+1}}\int_{I_{i,j+1/2}} (u,v)\cdot\vec{n}(s)q(x(s),y(s),t)dsdt. \tag{2.3b}$$

The simplest finite volume method for the advection equation is the first-order DCU method. Its fluxes are

$$\begin{aligned} F_{i+\frac{1}{2},j}^D &= (U_{i+\frac{1}{2},j}^0)^+ Q_{i,j}^n + (U_{i+\frac{1}{2},j}^0)^- Q_{i+1,j}^n, \\ G_{i,j+\frac{1}{2}}^D &= (V_{i,j+\frac{1}{2}}^0)^+ Q_{i,j}^n + (V_{i,j+\frac{1}{2}}^0)^- Q_{i,j+1}^n, \end{aligned} \tag{2.4}$$

where the superscript 'D' in the flux function expresses the DCU flux. If we denote advection velocity by $\vec{u}_{i+1/2,j}^A = (u_{i+1/2,j}, v_{i+1/2,j})$, $\vec{v}_{i,j+1/2}^A = (u_{i,j+1/2},\ v_{i,j+1/2})$, then the transport velocities at normal direction of edges $I_{i+1/2,j}$ and $I_{i,j+1/2}$ are

$$U_{i+\frac{1}{2},j}^0 = \vec{u}_{i+1/2,j}^A \cdot \vec{n}_{i+1/2,j}, \quad V_{i,j+1/2}^0 = \vec{v}_{i,j+1/2}\cdot\vec{n}_{i,j+1/2}, \tag{2.5}$$

where $\vec{n}_{i+1/2,j}$,$\vec{n}_{i,j+1/2}$ are outward normal directions of edges $I_{i+1/2,j}$ and $I_{i,j+1/2}$. In addition, $u^+ = \max(u,0),\ u^- = \min(u,0)$.

Colella [5] took account of the flow direction more fully and proposed a better scheme on rectangular grid: the CTU method. Fig. 2.2 displays a distinguish between DCU and CTU fluxes in the situation of $U_{i+\frac{1}{2},j}^0 > 0,\ V_{i,j-1/2}^0 > 0$. For the general velocities $U^0$ and $V^0$, the numerical scheme (2.2) with the DCU flux has five-point stencils at most. The CTU method may include nine-point stencils, thus it has better stability.

When extending the CTU scheme to a general quadrilateral grid, we can still use finite volume method in physical space to describe CTU fluxes. But a simpler method is using the coordinate transformation, which is easier in designing more complex schemes.

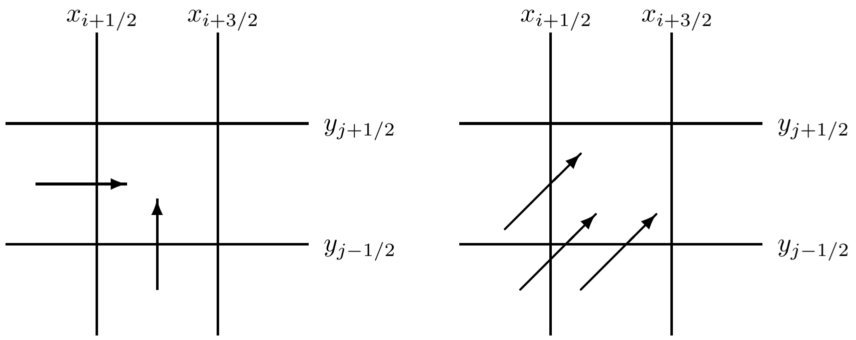

Fig. 2.2. Two advection ways in numerical algorithms for a physical quantity. (a) The DCU method. (b) The CTU method.

### 2.2. Coordinate transformation

In this subsection we will use coordinate transformation to introduce the CTU method. At first, a coordinate transformation for dealing with general domains is introduced, and the transformation maps an irregular domain in physical space $(x, y)$ to a perfectly regular computational domain in computational space $(\xi, \eta)$, see Fig. 2.3. The advection equation is discretized in a perfectly regular mesh as done for Cartesian domains. We know such methods may not be easy to exactly conserve the correct physical quantities in the physical space, and they have strict requirements to the smoothness of mapping transformation. To cure such defects, we construct a first-order discretization of the transformed equation and keep it be consistent with the DCU finite volume method (2.2) in the physical domain; then construct a CTU, a high-order accurate flux scheme for the transformation equation. Similar coordinate transformed idea has been used in more complicated flow problems [8].

Define the transformation via

$$\begin{cases} dx = Ad\xi + Ld\eta, \\ dy = Bd\xi + Md\eta, \end{cases}$$

where $K = (A, B, L, M)$ are the geometric variables satisfying

$$A = x_\xi, \quad L = x_\eta, \quad B = y_\xi, \quad M = y_\eta, \quad A_\eta = L_\xi, \quad B_\eta = M_\xi. \tag{2.6}$$

The advection equation (2.1) defined in physical space in terms of the $(x, y)$ coordinates is transformed into computational space in terms of the generalized coordinates $(\xi, \eta)$ as

$$(Jq)_t + \frac{\partial}{\partial \xi}(Uq) + \frac{\partial}{\partial \eta}(Vq) = 0, \tag{2.7}$$

where $J$ is the determinant of Jacobian transformation matrix, i.e. $J = AM - BL$. $U, V$ are transformed advection velocities,

$$U = Mu - Lv, \quad V = -Bu + Av.$$

The discrete scheme in computational space for advection equation (2.7) can be expressed by

$$Q_{ij}^{n+1} = Q_{ij}^n - \frac{\Delta t}{J_{ij}\Delta\xi}(F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}) - \frac{\Delta t}{J_{ij}\Delta\eta}(G_{i,j+\frac{1}{2}} - G_{i,j-\frac{1}{2}}), \tag{2.8}$$

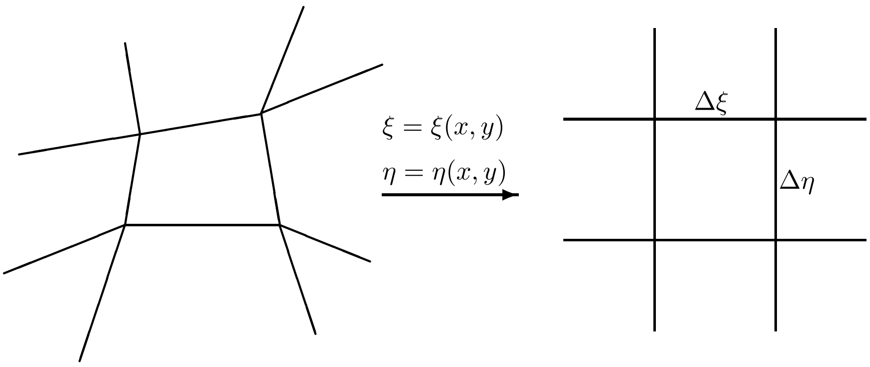

Fig. 2.3. The physical grid cells (left) are mapped to computational grid cells (right).

where

$$Q_{ij}^n=\frac{\int\int Jq(\xi,\eta,t^n)d\xi d\eta}{J_{ij}\Delta\xi\Delta\eta},$$
$$F_{i+\frac{1}{2},j}=\frac{1}{\Delta t\Delta\eta}\int_{t^n}^{t^{n+1}}\int_{\eta_{j-\frac{1}{2}}}^{\eta_{j+\frac{1}{2}}}Uqd\eta dt,\quad G_{i,j+\frac{1}{2}}=\frac{1}{\Delta t\Delta\xi}\int_{t^n}^{t^{n+1}}\int_{\xi_{i-\frac{1}{2}}}^{\xi_{i+\frac{1}{2}}}Vqd\xi dt,$$

$J_{ij}=|C_{ij}|/\Delta\xi\Delta\eta$, and $|C_{i,j}|$ is the area of the cell $C_{i,j}$. Note that $F_{i+1/2,j}$ is the flux per unit length in computational space, which is different from the one in (2.2) with a scaling factor.

In order to keep the consistency with the finite volume scheme (2.2) on arbitrary quadrilateral grids, there must be

$$\int_{I_{i+1/2,j}}(u,v)\cdot\vec{n}(s)qds=\int_{\eta_{j-\frac{1}{2}}}^{\eta_{j+\frac{1}{2}}}Uqd\eta,\quad \int_{I_{i,j+1/2}}(u,v)\cdot\vec{n}(s)qds=\int_{\xi_{i-\frac{1}{2}}}^{\xi_{i+\frac{1}{2}}}Vqd\xi.$$

To satisfy these conditions it is enough to let

$$\begin{aligned}M_{i+\frac{1}{2},j}&=\frac{y_{i+1/2,j+1/2}-y_{i+1/2,j-1/2}}{\Delta\eta},\quad L_{i+\frac{1}{2},j}=\frac{x_{i+1/2,j+1/2}-x_{i+1/2,j-1/2}}{\Delta\eta},\\ A_{i,j+\frac{1}{2}}&=\frac{x_{i+1/2,j+1/2}-x_{i-1/2,j+1/2}}{\Delta\xi},\quad B_{i,j+\frac{1}{2}}=\frac{y_{i+1/2,j+1/2}-y_{i-1/2,j+1/2}}{\Delta\xi},\end{aligned}\tag{2.9}$$

which satisfies the discrete Piola compatibility conditions (2.6). Denote

$$\begin{aligned}U_{i+\frac{1}{2},j}&=M_{i+\frac{1}{2},j}u_{i+\frac{1}{2},j}-L_{i+\frac{1}{2},j}v_{i+\frac{1}{2},j},\\ V_{i,j+\frac{1}{2}}&=-B_{i,j+\frac{1}{2}}u_{i,j+\frac{1}{2}}+A_{i,j+\frac{1}{2}}v_{i,j+\frac{1}{2}}.\end{aligned}\tag{2.10}$$

Then $U_{i+\frac{1}{2},j}$ and $V_{i,j+\frac{1}{2}}$ describe normal velocities at edges. The DCU fluxes are

$$F^D_{i+\frac{1}{2},j}=U^+_{i+\frac{1}{2},j}Q^n_{ij}+U^-_{i+\frac{1}{2},j}Q^n_{i+1,j},\quad G^D_{i,j+\frac{1}{2}}=V^+_{i,j+\frac{1}{2}}Q^n_{ij}+V^-_{i,j+\frac{1}{2}}Q^n_{i,j+1}.\tag{2.11}$$

For the CTU scheme, one may obtain flux by tracing back the characteristics $q(\xi-U\Delta t,\eta-V\Delta t,t^n)$, but a simple and unified derivation to DCU, CTU and high resolution approaches is to use Taylor series expansion.

We make a Taylor expansion

$$\begin{aligned} q(\xi,\eta,t^{n+1}) &= q(\xi,\eta,t^n)+\Delta t q_t+\frac{\Delta t^2}{2}q_{tt}+O(\Delta t^3)\\ &= q(\xi,\eta,t^n)-\frac{1}{J}\Delta t(Uq)_\xi-\frac{1}{J}\Delta t(Vq)_\eta \\ &\quad +\frac{\Delta t^2}{2J}\Big((\frac{U}{J}(Uq)_\xi)_\xi+(\frac{U}{J}(Vq)_\eta)_\xi+(\frac{V}{J}(Uq)_\xi)_\eta+(\frac{V}{J}(Vq)_\eta)_\eta\Big)+O(\Delta t^3). \end{aligned} \tag{2.12}$$

Obviously, the DCU fluxes (2.11) use upwind approximations to the derivatives $(Uq)_\xi/J$ and $(Vq)_\eta/J$ in the $O(\Delta t)$ terms of the above expansion.

The CTU flux modification will be provided by the discretization of cross-derivative terms $(U(Vq)_\eta/J)_\xi/J$ and $(V(Uq)_\xi/J)_\eta/J$ in the $O((\Delta t)^2)$ terms of the Taylor expansion (2.12), that is

$$\begin{aligned} F^C_{i+\frac{1}{2},j} &= F^D_{i+\frac{1}{2},j}-\frac{1}{2}\frac{\Delta t}{J_{i+1/2,j}\Delta\eta}\Big(U^+_{i+\frac{1}{2},j}(\mathcal{B}^+\Delta Q_{i,j-\frac{1}{2}}+\mathcal{B}^-\Delta Q_{i,j+\frac{1}{2}})\\ &\quad +U^-_{i+\frac{1}{2},j}(\mathcal{B}^+\Delta Q_{i+1,j-\frac{1}{2}}+\mathcal{B}^-\Delta Q_{i+1,j+\frac{1}{2}})\Big), \end{aligned} \tag{2.13a}$$

$$\begin{aligned} G^C_{i,j+\frac{1}{2}} &= G^D_{i,j+\frac{1}{2}}-\frac{1}{2}\frac{\Delta t}{J_{i,j+1/2}\Delta\xi}\Big(V^+_{i,j+\frac{1}{2}}(\mathcal{A}^+\Delta Q_{i-\frac{1}{2},j}+\mathcal{A}^-\Delta Q_{i+\frac{1}{2},j})\\ &\quad +V^-_{i,j+\frac{1}{2}}(\mathcal{A}^+\Delta Q_{i-\frac{1}{2},j+1}+\mathcal{A}^-\Delta Q_{i+\frac{1}{2},j+1})\Big), \end{aligned} \tag{2.13b}$$

where the superscript 'C' in the flux function expresses the CTU flux, $J_{ij}\Delta\xi$ in (2.8) represents average grid space in $\xi$ -direction within a grid cell $C_{ij}$, and $J_{i+1/2,j}\Delta\xi$ represents average grid space between two cells $C_{ij}$ and $C_{i+1,j}$. Thus $J_{i+1/2,j}$ and $J_{i,j+1/2}$ usually have forms as

$$J_{i+1/2,j}=(J_{i,j}+J_{i+1,j})/2,\quad J_{i,j+1/2}=(J_{i,j}+J_{i,j+1})/2.$$

Partial cross-derivative terms

$$\mathcal{B}^\pm\Delta Q_{i,j-\frac{1}{2}}=V^\pm_{i,j-\frac{1}{2}}(Q^n_{ij}-Q^n_{i,j-1}),\quad \mathcal{A}^\pm\Delta Q_{i+\frac{1}{2},j}=U^\pm_{i+\frac{1}{2},j}(Q^n_{i+1,j}-Q^n_{i,j}), \tag{2.14}$$

introduce transverse propagation of waves $Q^n_{ij}-Q^n_{i,j-1}$ and $Q^n_{i+1,j}-Q^n_{i,j}$ to fluxes $F_{i+\frac{1}{2},j}$ and $G_{i,j+\frac{1}{2}}$. Then the transverse jumps are transported by advection velocities $U$ and $V$. Fig. 2.4 gives the case of $F_{i+1/2,j}$. Normally three out of these four corrections will vanish, as in the case that $U$ and $V$ are constants. Due to the contribution of these transverse waves to fluxes, the corner information is introduced.

The CTU schemes fail to be second-order accurate because of lack of discretization to terms $(U(Uq)_\eta/J)_\xi$ and $(V(Vq)_\xi/J)_\eta$ in the $O((\Delta t)^2)$ terms of the Taylor expansion (2.12). These two modification terms are discretized in [18] as

$$\widetilde{F}_{i+\frac{1}{2},j}=\frac{1}{2}|U_{i+\frac{1}{2},j}|(1-|U_{i+\frac{1}{2},j}|\frac{\Delta t}{J_{i+1/2,j}\Delta\xi})\widetilde{W}_{i+\frac{1}{2},j}, \tag{2.15a}$$

$$\widetilde{G}_{i,j+\frac{1}{2}}=\frac{1}{2}|V_{i,j+\frac{1}{2}}|(1-|V_{i,j+\frac{1}{2}}|\frac{\Delta t}{J_{i,j+1/2}\Delta\eta})\widetilde{W}_{i,j+\frac{1}{2}}. \tag{2.15b}$$

where $\widetilde{W}_{i+1/2,j}$ is a limited version of wave $W_{i+1/2,j}=Q^n_{i+1,j}-Q^n_{i,j}$,

$$\widetilde{W}_{i+1/2,j}=\phi(\theta_{i+1/2,j})W_{i+1/2,j},$$

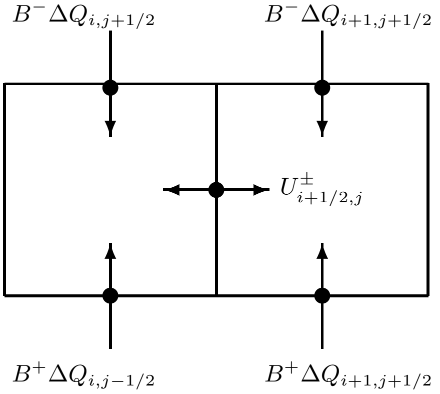

Fig. 2.4. The correction terms in the CTU flux $F_{i+1/2,j}$ consist of four possible transverse jumps. Their contributions to the flux depend on single-point velocities $U^{\pm}_{i+1/2,j}$.

where $\theta_{i+1/2,j}$ should be some measure of the smoothness of the single wave. It is easily obtained by comparing this jump with those at the neighboring Riemann problem in the upwind direction. Denote

$$I = \begin{cases} i, & \text{if } \ u_{i+1/2,j} > 0. \\ i+1, & \text{if } \ u_{i+1/2,j} < 0. \end{cases}$$

Then we have

$$\theta_{i+1/2,j} = \frac{W_{I+1/2,j}}{W_{i+1/2,j}}.$$

$\phi$ is a high-resolution limiter, for example, $\phi(\theta) = minmod(1, \theta)$ is the minmode limiter, and $\phi(\theta) = \max(0, \min((1+\theta)/2, 2, 2\theta)$ is the monotinized centered one. For more details see [18]. The similar expresses for $\widetilde{W}_{i,j+\frac{1}{2}}$ can be obtained.

The final fluxes including high-resolution correction terms can be implemented by adding corrections in the $\xi$- and $\eta$- directions separately, for example,

$$F^{D,2}_{i+\frac{1}{2},j} = F^{D}_{i+\frac{1}{2},j} + \widetilde{F}_{i+\frac{1}{2},j}, \quad G^{D,2}_{i,j+\frac{1}{2}} = G^{D}_{i,j+\frac{1}{2}} + \widetilde{G}_{i,j+\frac{1}{2}}, \tag{2.16}$$

$$F^{C,2}_{i+\frac{1}{2},j} = F^{C}_{i+\frac{1}{2},j} + \widetilde{F}_{i+\frac{1}{2},j}, \quad G^{C,2}_{i,j+\frac{1}{2}} = G^{C}_{i,j+\frac{1}{2}} + \widetilde{G}_{i,j+\frac{1}{2}}, \tag{2.17}$$

where the superscript '2' in a flux function denotes the resulting flux with second-order accurate modification.

During the computation, we introduce two constants $\Delta\xi$ and $\Delta\eta$, whose choices do not affect the computing results. In our code, we let $\Delta\xi = L_\xi/N_\xi, \Delta\eta = L_\eta/N_\eta$, where $L_\xi, L_\eta$ are domain lengths in $\xi$ and $\eta$ directions, and $N_\xi, N_\eta$ are cell numbers in these two directions.

**Remark 2.1.** The CTU method has better stability properties than DCU one. The stable steplength limit condition for the CTU method is

$$\Delta t \max\left(\frac{U^+_{i-1/2,j}}{J_{ij}\Delta\xi}, \frac{-U^-_{i+1/2,j}}{J_{ij}\Delta\xi}, \frac{V^+_{i,j-1/2}}{J_{ij}\Delta\eta}, \frac{-V^-_{i,j+1/2}}{J_{ij}\Delta\eta}\right) \le C_{cfl},$$

where $C_{cfl} < 1$ is Courant number. The stable condition for DCU method is

$$\Delta t\left(\frac{U^+_{i-1/2,j}}{J_{ij}\Delta\xi} + \frac{-U^-_{i+1/2,j}}{J_{ij}\Delta\xi} + \frac{V^+_{i,j-1/2}}{J_{ij}\Delta\eta} + \frac{-V^-_{i,j+1/2}}{J_{ij}\Delta\eta}\right) \le C_{cfl}.$$

**Remark 2.2.** The discretization scheme (2.8) of the transformed equation (2.7) is consistent with a finite volume method (2.2) in the meaning of keeping the same DCU algorithm, so the method remains valid and accurate even on a highly nonuniform grid (corresponding to a nonsmooth mapping). At the same time, the conservation of quantity $q(x,y,t)$ is strictly preserved in physical space.

## 3. A Modified Advection Algorithm

When advection velocity $\vec{u}^A=(u,v)$ varies smoothly along a cell edge $I_{i+1/2,j}$, using single-edge velocity approximation $(u_{i+1/2,j},v_{i+1/2,j})$ can provide good results (Fig. 3.1(a)). But if transverse variation of advection velocity is very large, such as in Fig. 3.1(b), single-edge velocity transportation may result in large errors. Here we describe a new multi-point flux method which includes more information.

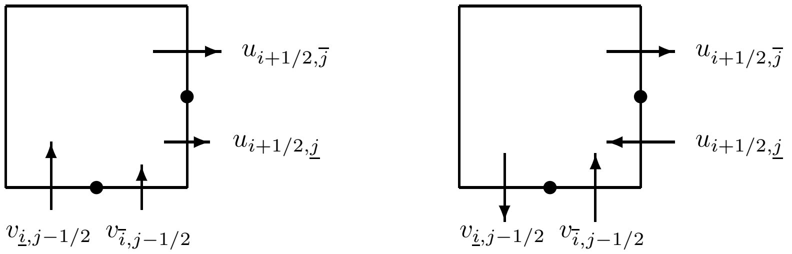
$u_{i+1/2,\bar{j}}$
$u_{i+1/2,\underline{j}}$
$v_{\underline{i},j-1/2}$
$v_{\bar{i},j-1/2}$
$u_{i+1/2,\bar{j}}$
$u_{i+1/2,\underline{j}}$
$v_{\underline{i},j-1/2}$
$v_{\bar{i},j-1/2}$

Fig. 3.1. Normal velocity distribution along an edge of a grid cell. (a) Small variation. (b) Large variation.

Introduce two advection velocities $\vec{u}^A_{i+1/2,\underline{j}}$ and $\vec{u}^A_{i+1/2,\bar{j}}$ at each cell edge $I_{i+1/2,j}$, where $\vec{u}^A_{i+1/2,\underline{j}}=(u_{i+1/2,\underline{j}},v_{i+1/2,\underline{j}})$ and $\vec{u}^A_{i+1/2,\bar{j}}=(u_{i+1/2,\bar{j}},v_{i+1/2,\bar{j}})$ represent the velocities at the lower and upper half-face of $I_{i+1/2,j}$, respectively. Similarly, $\vec{u}^A_{\underline{i},j+1/2}$ and $\vec{u}^A_{\bar{i},j+1/2}$ are half-face velocities at edge $I_{i,j+1/2}$ (Fig. 3.1). Then the flux of physical quantity $q$ passing $I_{i+1/2,j}$ will have multi-point characteristic.

Adopting similar derivative procedure in Section 2.2, the multi-point flux algorithm on quadrilateral grid with variable coefficients can be constructed easily. Let us write down conclusions to those schemes.

In the DCU method, the edge flux is the summation of partial fluxes

$$F^{D,M}_{i+\frac{1}{2},j}=\frac{1}{2}\Big(U^{+}_{i+\frac{1}{2},\underline{j}}+U^{+}_{i+\frac{1}{2},\bar{j}}\Big)Q^n_{ij}+\frac{1}{2}\Big(U^{-}_{i+\frac{1}{2},\underline{j}}+U^{-}_{i+\frac{1}{2},\bar{j}}\Big)Q^n_{i+1,j}, \tag{3.1a}$$

$$G^{D,M}_{i,j+\frac{1}{2}}=\frac{1}{2}\Big(V^{+}_{\underline{i},j+\frac{1}{2}}+V^{+}_{\bar{i},j+\frac{1}{2}}\Big)Q^n_{ij}+\frac{1}{2}\Big(V^{-}_{\underline{i},j+\frac{1}{2}}+V^{-}_{\bar{i},j+\frac{1}{2}}\Big)Q^n_{i,j+1}, \tag{3.1b}$$

where the superscript 'M' means multi-point flux, and

$$U_{i+\frac{1}{2},\underline{j}}=M_{i+\frac{1}{2},j}u_{i+\frac{1}{2},\underline{j}}-L_{i+\frac{1}{2},j}v_{i+\frac{1}{2},\underline{j}},\qquad U_{i+\frac{1}{2},\bar{j}}=M_{i+\frac{1}{2},j}u_{i+\frac{1}{2},\bar{j}}-L_{i+\frac{1}{2},j}v_{i+\frac{1}{2},\bar{j}},$$

$$V_{\underline{i},j+\frac{1}{2}}=-B_{i,j+\frac{1}{2}}u_{\underline{i},j+\frac{1}{2}}+A_{i,j+\frac{1}{2}}v_{\underline{i},j+\frac{1}{2}},\qquad V_{\bar{i},j+\frac{1}{2}}=-B_{i,j+\frac{1}{2}}u_{\bar{i},j+\frac{1}{2}}+A_{i,j+\frac{1}{2}}v_{\bar{i},j+\frac{1}{2}},$$

where geometric variables $A,B,L,M$ are the same as given in (2.9).

Note that if velocities $U_{i+\frac{1}{2},\underline{j}}$ and $U_{i+\frac{1}{2},\bar{j}}$ have same signs, and $V_{\underline{i},j+\frac{1}{2}}$ and $V_{\bar{i},j+\frac{1}{2}}$ have same signs, then flux (3.1) degenerates to the classical DCU fluxes (2.11).

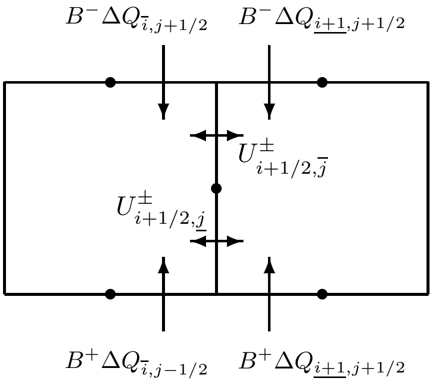
$B^-\Delta Q_{\bar{i},j+1/2}$
$B^-\Delta Q_{\underline{i+1},j+1/2}$
$U^{\pm}_{i+1/2,\bar{j}}$
$U^{\pm}_{i+1/2,\underline{j}}$
$B^+\Delta Q_{\bar{i},j-1/2}$
$B^+\Delta Q_{\underline{i+1},j+1/2}$

Fig. 3.2. The correction terms in the multi-point CTU flux $F_{i+1/2,j}$ consist of four local transverse jumps. Their contributions to the flux $F_{i+1/2,j}$ depend on local velocities.

For CTU scheme, there are

$$F^{C,M}_{i+\frac{1}{2},j} = F^{D,M}_{i+\frac{1}{2},j} - \frac{1}{2}\frac{\Delta t}{J_{i+1/2,j}\Delta\eta}\Big(U^+_{i+\frac{1}{2},\underline{j}}\mathcal{B}^+\Delta Q_{\bar{i},j-\frac{1}{2}} + U^+_{i+\frac{1}{2},\bar{j}}\mathcal{B}^-\Delta Q_{\bar{i},j+\frac{1}{2}} + U^-_{i+\frac{1}{2},\underline{j}}\mathcal{B}^+\Delta Q_{\underline{i+1},j-\frac{1}{2}} + U^-_{i+\frac{1}{2},\bar{j}}\mathcal{B}^-\Delta Q_{\underline{i+1},j+\frac{1}{2}}\Big), \tag{3.2}$$

$$G^{C,M}_{i,j+\frac{1}{2}} = G^{D,M}_{i,j+\frac{1}{2}} - \frac{1}{2}\frac{\Delta t}{J_{i,j+1/2}\Delta\xi}\Big(V^+_{\underline{i},j+\frac{1}{2}}\mathcal{A}^+\Delta Q_{i-\frac{1}{2},\bar{j}} + V^+_{\bar{i},j+\frac{1}{2}}\mathcal{A}^-\Delta Q_{i+\frac{1}{2},\bar{j}} + V^-_{\underline{i},j+\frac{1}{2}}\mathcal{A}^+\Delta Q_{i-\frac{1}{2},\underline{j+1}} + V^-_{\bar{i},j+\frac{1}{2}}\mathcal{A}^-\Delta Q_{i+\frac{1}{2},\underline{j+1}}\Big), \tag{3.3}$$

where

$$\begin{aligned} &\mathcal{B}^{\pm}\Delta Q_{\underline{i},j-\frac{1}{2}} = V^{\pm}_{\underline{i},j-\frac{1}{2}}(Q^n_{ij} - Q^n_{i,j-1}), \qquad \mathcal{B}^{\pm}\Delta Q_{\bar{i},j-\frac{1}{2}} = V^{\pm}_{\bar{i},j-\frac{1}{2}}(Q^n_{ij} - Q^n_{i,j-1}),\\ &\mathcal{A}^{\pm}\Delta Q_{i+\frac{1}{2},\underline{j}} = U^{\pm}_{i+\frac{1}{2},\underline{j}}(Q^n_{i+1,j} - Q^n_{i,j}), \qquad \mathcal{A}^{\pm}\Delta Q_{i+\frac{1}{2},\bar{j}} = U^{\pm}_{i+\frac{1}{2},\bar{j}}(Q^n_{i+1,j} - Q^n_{i,j}). \end{aligned} \tag{3.4}$$

The discrepancies between new multi-point CTU fluxes (3.2)-(3.3) and classical ones (2.13) can be illustrated by comparing Fig. 2.4 and Fig. 3.2. The corrections in the traditional CTU flux $F_{i+1/2,j}$ consist of four possible global transverse jumps. Whereas the corrections in the new multi-point CTU flux $F_{i+1/2,j}$ consist of four possible local transverse jumps. The contributions of these local jumps depend on local propagation velocities $U^{\pm}_{i+1/2,\underline{j}}$ and $U^{\pm}_{i+1/2,\bar{j}}$.

The second-order accurate numerical fluxes in (2.8) are

$$\begin{aligned} F^{C,M,2}_{i+\frac{1}{2},j} &= F^{C,M}_{i+\frac{1}{2},j} + \frac{1}{2}(\widetilde{F}_{i+\frac{1}{2},\underline{j}} + \widetilde{F}_{i+\frac{1}{2},\bar{j}}),\\ G^{C,M,2}_{i,j+\frac{1}{2}} &= G^{C,M}_{i,j+\frac{1}{2}} + \frac{1}{2}(\widetilde{G}_{\underline{i},j+\frac{1}{2}} + \widetilde{G}_{\bar{i},j+\frac{1}{2}}), \end{aligned}$$

where $\widetilde{F}_{i+\frac{1}{2},\underline{j}}$ represents high-order modification terms (2.15) with $U_{i+\frac{1}{2},j}$ replaced by $U_{i+\frac{1}{2},\underline{j}}$. The other expression has the similar transformation way.

The stable steplength limit condition for the CTU method is

$$\Delta t \max\left(\frac{U^+_{i-1/2,\underline{j}}}{J_{ij}\Delta\xi}, \frac{U^+_{i-1/2,\bar{j}}}{J_{ij}\Delta\xi}, \frac{-U^-_{i+1/2,\underline{j}}}{J_{ij}\Delta\xi}, \frac{-U^-_{i+1/2,\bar{j}}}{J_{ij}\Delta\xi}, \frac{V^+_{\underline{i},j-1/2}}{J_{ij}\Delta\eta}, \frac{V^+_{\bar{i},j-1/2}}{J_{ij}\Delta\eta}, \frac{-V^-_{\underline{i},j+1/2}}{J_{ij}\Delta\eta}, \frac{-V^-_{\bar{i},j+1/2}}{J_{ij}\Delta\eta}\right) \le C_{cfl}, \tag{3.5}$$

where Courant number $C_{cfl} < 1$.

**Remark 3.1.** If a grid can be refined arbitrarily, the multi-point flux method seems not to exhibit more advantages than the classical advection algorithm. But it is suitable to the following problems:

1. A grid is not allowed to be refined arbitrarily.
2. Even if a grid is refined, the transverse variation of the transport velocities at edge of grid cell still may have large variation.

The remapping problem belongs to such kind of ones.

## 4. The Remapping Method

### 4.1. Statement of the Remapping problem

The Remapping problem includes two grids and corresponding physical quantities on these grids. For the sake of consistency with latter discussion, we adopt the following notations. The physical quantity $q$ on the old grid has superscript $n$, whereas superscript $n+1$ indicates the quantity on the new (remapped) grid. For the quadrilateral grid, an old grid $C^n_{i,j}$ has coordinates vertices $P^n_{i-1/2,j-1/2}$, $P^n_{i+1/2,j-1/2}$, $P^n_{i+1/2,j+1/2}$ and $P^n_{i-1/2,j+1/2}$. The interface connecting vertices $P^n_{i+1/2,j-1/2}$ and $P^n_{i+1/2,j+1/2}$ is denoted by $I^n_{i+1/2,j}$, and its length is $|I^n_{i+1/2,j}|$. Similar notations will be employed in new grid with the superscript $n$ replaced by $n+1$, see Fig. 4.1(a).

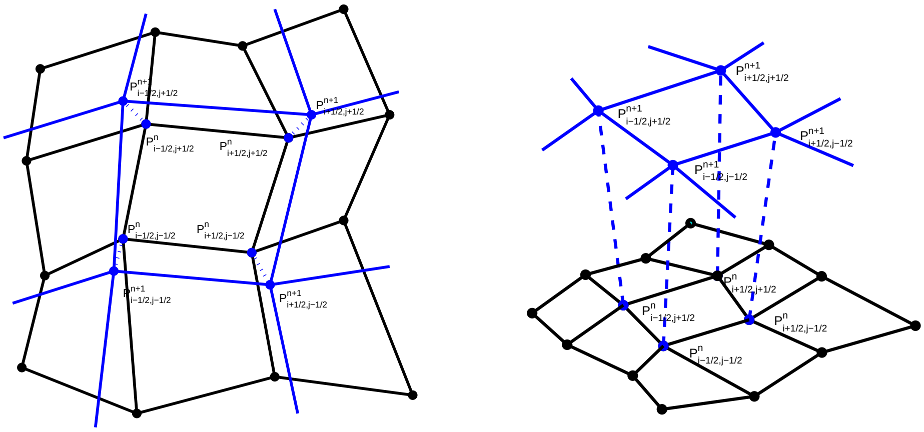

Fig. 4.1. Old and new mesh in remapping problem. (a) The planform plot.(b) The control volume with a pseudo time.

We assume that there is a function $q(x,y)$ defined on the domain. The only information available of this function is its mean value in each of the cells of the old grid:

$$Q^n_{i,j} = \frac{\int_{C^n_{i,j}} q(x,y)dxdy}{|C^n_{i,j}|}. \tag{4.1}$$

Once given a new grid, the problem statement is to find approximations to

$$Q^{n+1}_{i,j} = \frac{\int_{C^{n+1}_{i,j}} q(x,y)dxdy}{|C^{n+1}_{i,j}|},$$

and the new quantities should satisfy the following conservation requirement

$$\sum_{i,j} Q_{i,j}^{n+1}|C_{i,j}^{n+1}| = \sum_{i,j} Q_{i,j}^{n}|C_{i,j}^{n}|.$$

### 4.2. The remapping equation

Introducing a pseudo time $t$, then the continuous remapping problem can be regarded as being equivalent to the following equation (see, e.g., [31] [33])

$$\frac{\partial q}{\partial t} = 0 \tag{4.2}$$

with a grid moving from $C_{i,j}^n$ to $C_{i,j}^{n+1}$, see Fig. 4.1(b).

In fact, the remapping problem arises from a time splitting procedure during a time step when solving fluid hydrodynamics equations on a moving grid. In the time step, the mass, momentum and total energy keep constant but the grid moves. The conservative quantities naturally satisfy equation (4.2). The benefit of introducing remapping equation is to transform a conservation interpolation problem into solving a partial differential problem.

Let us observe two pictures: an initial physical distribution $q(x,y,0)$ is at rest where a grid moves with velocity $(-u,-v)$ or $q(x,y,0)$ shifts with velocity $(u,v)$ on a fixed grid. The former is described by equation (4.2) with grid velocity $(-u,-v)$ and the latter is by advection equation (2.1) with transport velocity $(u,v)$. Obviously the two physical phenomena are the same if the observer moves with grid.

### 4.3. The finite volume approach

It is convenient to recast equations (4.2) into the more fundamental control volume formulation, which holds for an arbitrary moving control volume:

$$\frac{d}{dt}\int_{C(x,y,t)} q dv = \int_{\partial C(x,y,t)} (u,v)\cdot \vec{n}(s(t)) q(x(s(t),t), y(s(t),t)ds,$$

where $C(x,y,t)$ is a control volume, $\partial C(x,y,t)$ is the boundary of $C(x,y,t)$, $\vec{n}(s(t))$ is the unit outward normal vector, and $s$ is the length element on $\partial C$. The boundary of the control volume is assumed to move with an arbitrary local velocity $(u,v)$.

In our notation, the moving velocity at vertex $P_{i+1/2,j+1/2}^n$ of grid cell $C_{i,j}(t)$ is

$$\vec{u}^M(P_{i+1/2,j+1/2}) = (u_{i+1/2,j+1/2}^{n+1/2},\ v_{i+1/2,j+1/2}^{n+1/2}),$$

where

$$u_{i+1/2,j+1/2}^{n+1/2} = \frac{x_{i+1/2,j+1/2}^{n+1} - x_{i+1/2,j+1/2}^{n}}{\Delta t}, \quad v_{i+1/2,j+1/2}^{n+1/2} = \frac{y_{i+1/2,j+1/2}^{n+1} - y_{i+1/2,j+1/2}^{n}}{\Delta t}.$$

Thus the edge $I_{i+1/2,j}(t)$ moves with

$$\vec{u}_{i+1/2,j}^M = 0.5\Big(\vec{u}^M(P_{i+1/2,j-1/2}) + \vec{u}^M(P_{i+1/2,j+1/2})\Big).$$

The fully discrete finite volume method for (4.2) in flux form is

$$\begin{aligned} Q_{ij}^{n+1} = \frac{|C_{i,j}^n|}{|C_{i,j}^{n+1}|}Q_{ij}^n + \frac{\Delta t}{|C_{i,j}^{n+1}|}\Big(&F_{i+\frac12,j}|I_{i+\frac12,j}^{n+1/2}| - F_{i-\frac12,j}|I_{i-\frac12,j}^{n+1/2}| \\ &+G_{i,j+\frac12}|I_{i,j+\frac12}^{n+1/2}| - G_{i,j-\frac12}|I_{i,j-\frac12}^{n+1/2}|\Big), \end{aligned} \tag{4.3}$$

where $Q^n_{i,j}$ is given by (4.1), and the fluxes are

$$F_{i+\frac{1}{2},j} = \frac{1}{\Delta t|I^{n+1/2}_{i+\frac{1}{2},j}|}\int_{t^n}^{t^{n+1}}\int_{I_{i+\frac{1}{2},j}(t)}(u,v)\cdot\vec{n}(s(t))q(x(s(t),t),y(s(t),t)dsdt, \tag{4.4a}$$

$$G_{i,j+\frac{1}{2}} = \frac{1}{\Delta t|I^{n+1/2}_{i,+\frac{1}{2}}|}\int_{t^n}^{t^{n+1}}\int_{I_{i,j+\frac{1}{2}}(t)}(u,v)\cdot\vec{n}(s(t))q(x(s(t),t),y(s(t),t)dsdt, \tag{4.4b}$$

where $|I^{n+1/2}_{i+\frac{1}{2},j}|$ is the average length of the interface $I_{i+\frac{1}{2},j}(t)$, that is

$$|I^{n+1/2}_{i+\frac{1}{2},j}| = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}}\int_{I_{i+\frac{1}{2},j}(t)}(u,v)\cdot\vec{n}(s(t))dsdt.$$

Substituting the grid moving velocity into $I_{i+\frac{1}{2},j}(t)$ gives

$$|I^{n+1/2}_{i+\frac{1}{2},j}| = \frac{1}{2}\Big(|I^n_{i+\frac{1}{2},j}| + |I^{n+1}_{i+\frac{1}{2},j}|\Big).$$

The time integration between $t^n$ and $t^{n+1}$ of the equations (4.4) raises the issue of where to evaluate the numerical flux function: on the mesh configuration at $n$ time layer, or on that at $n+1$ time layer, or in between these two configurations? This problem is often referred to as the geometric conservation law (GCL), which may be regarded as an identity that must be satisfied, if the conservative property is to be maintained [10] [27] [13] [17] [37] [38]. In the present situation the GCL is satisfied if fluxes are computed on the control volume at $t^{n+1/2}$.

Denote $\vec{n}^{n+1/2}_{i+1/2,j} = (\vec{n}^n_{i+1/2,j} + \vec{n}^{n+1}_{i+1/2,j})/2$ as the unit vector normal to the middle interface $I^{n+1/2}_{i+1/2,j}$, the fluxes in the DCU method are

$$F^D_{i+\frac{1}{2},j} = (U^0_{i+\frac{1}{2},j})^+Q^n_{i,j} + (U^0_{i+\frac{1}{2},j})^-Q^n_{i+1,j},$$
$$G^D_{i,j+\frac{1}{2}} = (V^0_{i,j+\frac{1}{2}})^+Q^n_{i,j} + (V^0_{i,j+\frac{1}{2}})^-Q^n_{i,j+1},$$

where $U^0_{i+1/2,j} = -\vec{u}^M_{i+1/2,j}\cdot\vec{n}^{n+1/2}_{i+1/2,j}$, $\quad V^0_{i,j+1/2} = -\vec{u}^M_{i,j+1/2}\cdot\vec{n}^{n+1/2}_{i,j+1/2}$.

Notice that the numerical fluxes of the remapping equation in the DCU method are the same as (2.4) of the advection equation except $U^0$ and $V^0$ with opposite signs.

**Remark 4.1.** Geometric conservation law states a conservation property between geometric quantities. When $q = 1$, note that

$$\Delta t F_{i+1/2,j}|I^{n+1/2}_{i+1/2,j}| = \Delta t U^0_{i+1/2,j}|I^{n+1/2}_{i+1/2,j}| = V_{i+1/2,j},$$

here $V_{i+1/2,j}$ is surface integrals over the region swept by the displacement of the cell faces from their old to their new locations $P^n_{i+1/2,j-1/2}P^n_{i+1/2,j+1/2}\ P^{n+1}_{i+1/2,j+1/2}P^{n+1}_{i+1/2,j-1/2}$, see Fig. 4.1(a). Similar equalities hold to other edges. From scheme (4.3) there is

$$|C^{n+1}_{i,j}| = |C^n_{i,j}| + V_{i+1/2,j} + V_{i-1/2,j} + V_{i,j+1/2} + V_{i,j-1/2}. \tag{4.5}$$

Thus, the discrete geometric conservation law associated with the numerical scheme (4.3) is satisfied.

### 4.4. The coordinate transformation approach

Once more we use coordinate transformation to introduce the CTU and more complicated schemes. We will derive finite volume methods in the physical domain, then view it as a discretization of the transformed equation. That means the discretization of the transformed equation is consistent with the moving mesh method.

Transform the independent variables in physical space $(x, y, t)$ to a new set of independent variables in a transformed space $(\xi, \eta, \tau)$ by

$$\begin{cases} dt = d\tau, \\ dx = ud\tau + Ad\xi + Ld\eta, \\ dy = vd\tau + Bd\xi + Md\eta. \end{cases} \tag{4.6}$$

Denote $K = (A, B, L, M)$ as geometric variables, and $W = (u, v)$ as the mesh moving velocity. The consistent relations between each two geometric variables are

$$A_\tau = u_\xi, \quad L_\tau = u_\eta, \quad A_\eta = L_\xi,$$

$$B_\tau = v_\xi, \quad M_\tau = v_\eta, \quad B_\eta = M_\xi.$$

Under the transformation (4.6), the remapping equation is transformed into

$$(Jq)_\tau + \frac{\partial}{\partial \xi}[(-Mu + Lv)q] + \frac{\partial}{\partial \eta}[(Bu - Av)q] = 0, \tag{4.7}$$

where $J = AM - BL$ is the determinant of Jacobian transformation matrix, which satisfies geometric conservation law

$$J_\tau + \frac{\partial}{\partial \xi}(-Mu + Lv) + \frac{\partial}{\partial \eta}(Bu - Av) = 0. \tag{4.8}$$

The remapping equation (4.7) has been transformed into an advection equation in a computational space. The advection algorithms in Section 2 and 3 can be used. But two new requirements need to be considered. One is the metric parameter $J$ varying with time $t$; the other is that the discretization of fluxes should satisfy discrete geometric conservation law. That means the discretization of (4.7) and (4.8) is consistent.

Denote $U = -Mu + Lv,\ V = Bu - Av$. The discrete scheme in computational space for the remapping equation (4.7) is

$$Q_{ij}^{n+1} = \frac{J_{ij}^n}{J_{ij}^{n+1}} Q_{ij}^n - \frac{\Delta t}{J_{ij}^{n+1}\Delta\xi}(F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}) - \frac{\Delta t}{J_{ij}^{n+1}\Delta\eta}(G_{i,j+\frac{1}{2}} - G_{i,j-\frac{1}{2}}), \tag{4.9}$$

where

$$F_{i+\frac{1}{2},j} = \frac{1}{\Delta t \Delta\eta}\int_{t^n}^{t^{n+1}}\int_{\eta_{j-\frac{1}{2}}}^{\eta_{j+\frac{1}{2}}} Uqd\eta dt, \quad G_{i,j+\frac{1}{2}} = \frac{1}{\Delta t \Delta\xi}\int_{t^n}^{t^{n+1}}\int_{\xi_{i-\frac{1}{2}}}^{\xi_{i+\frac{1}{2}}} Vqd\xi dt,$$

$$J_{ij}^n = |C_{ij}^n|/\Delta\xi\Delta\eta, \quad J_{ij}^{n+1} = |C_{ij}^{n+1}|/\Delta\xi\Delta\eta.$$

In order to keep the consistency with the finite volume scheme (4.3) on a moving quadrilateral grid, it is enough to let

$$M^{n+1/2}_{i+\frac{1}{2},j}=\frac{y^{n+1/2}_{i+1/2,j+1/2}-y^{n+1/2}_{i+1/2,j-1/2}}{\Delta\eta},\quad L^{n+1/2}_{i+\frac{1}{2},j}=\frac{x^{n+1/2}_{i+1/2,j+1/2}-x^{n+1/2}_{i+1/2,j-1/2}}{\Delta\eta},$$

$$A^{n+1/2}_{i,j+\frac{1}{2}}=\frac{x^{n+1/2}_{i+1/2,j+1/2}-x^{n+1/2}_{i-1/2,j+1/2}}{\Delta\xi},\quad B^{n+1/2}_{i,j+\frac{1}{2}}=\frac{y^{n+1/2}_{i+1/2,j+1/2}-y^{n+1/2}_{i-1/2,j+1/2}}{\Delta\xi},$$

$$u^{n+1/2}_{i+\frac{1}{2},j}=\frac{x^{n+1}_{i+1/2,j}-x^{n}_{i+1/2,j}}{\Delta t},\qquad v^{n+1/2}_{i+\frac{1}{2},j}=\frac{y^{n+1}_{i+1/2,j}-y^{n}_{i+1/2,j}}{\Delta t},$$

where $(x_{i+1/2,j}, y_{i+1/2,j})$ are the coordinates of the middle point $P_{i+1/2,j}$ at edge $I_{i+1/2,j}$.

Define

$$\begin{aligned}U_{i+\frac{1}{2},j}&=-M^{n+1/2}_{i+\frac{1}{2},j}u^{n+1/2}_{i+\frac{1}{2},j}+L^{n+1/2}_{i+\frac{1}{2},j}v^{n+1/2}_{i+\frac{1}{2},j},\\ V_{i,j+\frac{1}{2}}&=B^{n+1/2}_{i,j+\frac{1}{2}}u^{n+1/2}_{i,j+\frac{1}{2}}-A^{n+1/2}_{i,j+\frac{1}{2}}v^{n+1/2}_{i,j+\frac{1}{2}},\end{aligned}\tag{4.10}$$

as normal velocities on edges $I^{n+1/2}_{i+1/2,j}$ and $I^{n+1/2}_{i,j+1/2}$. The DCU fluxes are

$$F^D_{i+\frac{1}{2},j}=U^+_{i+\frac{1}{2},j}Q^n_{i,j}+U^-_{i+\frac{1}{2},j}Q^n_{i+1,j},\quad G^D_{i,j+\frac{1}{2}}=V^+_{i,j+\frac{1}{2}}Q^n_{i,j}+V^-_{i,j+\frac{1}{2}}Q^n_{i,j+1}.\tag{4.11}$$

The two fluxes have the same form as (2.11) except the definition of $U$ and $V$ in (2.10) is now replaced by (4.10). It is noted from (4.10) that

$$U_{i+\frac{1}{2},j}=|I^{n+1/2}_{i+1/2,j}|/\Delta\eta U^0_{i+\frac{1}{2},j},\ \ V_{i,j+\frac{1}{2}}=|I^{n+1/2}_{i,j+1/2}|/\Delta\xi V^0_{i,j+\frac{1}{2}}.$$

The DCU scheme (4.9) is then equivalent to (4.3).

### 4.4.1. The CTU method and high-order corrections

Construction of the CTU fluxes and high-order corrections depends on Taylor expansion

$$(Jq)(\xi,\eta,\tau^{n+1})=(Jq)(\xi,\eta,\tau^n)+\Delta t(Jq)_\tau+\frac{1}{2}\Delta t^2(Jq)_{\tau\tau}+O(\Delta t^3),\tag{4.12}$$

where from (4.7), there is

$$(Jq)_\tau=-(Uq)_\xi-(Vq)_\eta,\tag{4.13}$$

and

$$\begin{aligned}(Jq)_{\tau\tau}&=-(Uq)_{\xi\tau}-(Vq)_{\eta\tau}=-(U_\tau q)_\xi-(V_\tau q)_\eta-(Uq_\tau)_\xi-(Vq_\tau)_\eta\\ &=-(U_\tau q)_\xi-(V_\tau q)_\eta+\Big(\frac{J_\tau}{J}Uq\Big)_\xi+\Big(\frac{J_\tau}{J}Vq\Big)_\eta\\ &\quad+\Big(\frac{U}{J}(Uq)_\xi\Big)_\xi+\Big(\frac{U}{J}(Vq)_\eta\Big)_\xi+\Big(\frac{V}{J}(Uq)_\xi\Big)_\eta+\Big(\frac{V}{J}(Vq)_\eta\Big)_\eta.\end{aligned}\tag{4.14}$$

Substitute (4.13) and (4.14) into (4.12), let us consider discrete form of each term in this expansion. The CTU fluxes include cross-derivative terms $(U(Vq)_\eta/J)_\xi$ and $(V(Uq)_\xi/J)_\eta$, thus

$$\begin{aligned}F^C_{i+\frac{1}{2},j}=F^D_{i+\frac{1}{2},j}-\frac{1}{2}\frac{\Delta t}{J^{n+1}_{i+1/2,j}\Delta\eta}&(U^+_{i+\frac{1}{2},j}(\mathcal{B}^+\Delta Q_{i,j-\frac{1}{2}}+\mathcal{B}^-\Delta Q_{i,j+\frac{1}{2}})\\ &+U^-_{i+\frac{1}{2},j}(\mathcal{B}^+\Delta Q_{i+1,j-\frac{1}{2}}+\mathcal{B}^-\Delta Q_{i+1,j+\frac{1}{2}})),\end{aligned}$$

$$G^C_{i,j+\frac{1}{2}} = G^D_{i,j+\frac{1}{2}} - \frac{1}{2}\frac{\Delta t}{J^{n+1}_{i,j+1/2}\Delta\xi}(V^+_{i,j+\frac{1}{2}}(\mathcal{A}^+\Delta Q_{i-\frac{1}{2},j} + \mathcal{A}^-\Delta Q_{i+\frac{1}{2},j})$$

$$+ V^-_{i,j+\frac{1}{2}}(\mathcal{A}^+\Delta Q_{i-\frac{1}{2},j+1} + \mathcal{A}^-\Delta Q_{i+\frac{1}{2},j+1})),$$

where $\mathcal{A}^\pm$ and $\mathcal{B}^\pm$ have the same expressions as (2.14) except the definition of $U$ and $V$.

The high-order corrections are

$$\widetilde{F}_{i+\frac{1}{2},j} = \frac{1}{2}|U_{i+\frac{1}{2},j}|(1 - |U_{i+\frac{1}{2},j}|\frac{\Delta t}{J^{n+1}_{i+1/2,j}\Delta\xi})\widetilde{W}_{i+\frac{1}{2},j},$$

$$\widetilde{G}_{i,j+\frac{1}{2}} = \frac{1}{2}|V_{i,j+\frac{1}{2}}|(1 - |V_{i,j+\frac{1}{2}}|\frac{\Delta t}{J^{n+1}_{i,j+1/2}\Delta\eta})\widetilde{W}_{i,j+\frac{1}{2}}.$$

Jacobi $J$ in these fluxes may be more properly evaluated at time $t^{n+1/2}$ rather than $t^{n+1}$, however, if the mesh varies smoothly this only affects high-order terms.

**Remark 4.2.** Neglecting other second order terms in (4.14), the flux form in remapping method is the same as that in advection equation except definition of $U$, $V$ and $J$.

Table 4.1: Comparisons of the advection algorithm and the remapping one.

| | Advection algorithm | Remapping algorithm |
|---|---|---|
| Numerical schemes | (2.8) | (4.9) |
| $U_{i+1/2,j}$, $V_{i,j+1/2}$ | (2.10) | (4.10) |
| Space steplength in $F_{i+1/2,j}$ | $J_{i+1/2,j}\Delta\xi$, $J_{i+1/2,j}\Delta\eta$ | $J^{n+1}_{i+1/2,j}\Delta\xi$, $J^{n+1}_{i+1/2,j}\Delta\eta$ |
| Space steplength in $G_{i+1/2,j}$ | $J_{i,j+1/2}\Delta\xi$, $J_{i,j+1/2}\Delta\eta$ | $J^{n+1}_{i,j+1/2}\Delta\xi$, $J^{n+1}_{i,j+1/2}\Delta\eta$ |

The numerical results are good enough using the above remapping scheme, so we would not make further discretization to other terms. In general, to obtain an overall second-order accurate scheme in moving mesh frame is rather difficult, not only due to poor accuracy at extrema (local maxima or minima in q), but also due to the discretization to terms $U^\pm_\tau$ and $V^\pm_\tau$.

### 4.4.2. The multi-point flux CTU method

In continuous remapping computation, possible relative locations of old and new meshes are shown in Fig. 4.2(a) and 4.2(b). In the situation depicted in Fig. 4.2(a), the moving directions at the vertexes $P_{i-1/2,j-1/2}$ and $P_{i-1/2,j+1/2}$ coincide with edge normal direction, so the single-edge flux is enough. But for the case in Fig. 4.2(b), the volumes of the total swept regions at $I_{i+1/2,j}$ may approximate to zero due to the fact that two opposite direction vertex velocities may cancel out, however, when physical quantity has large difference between edge $I_{i-1/2,j}$, the

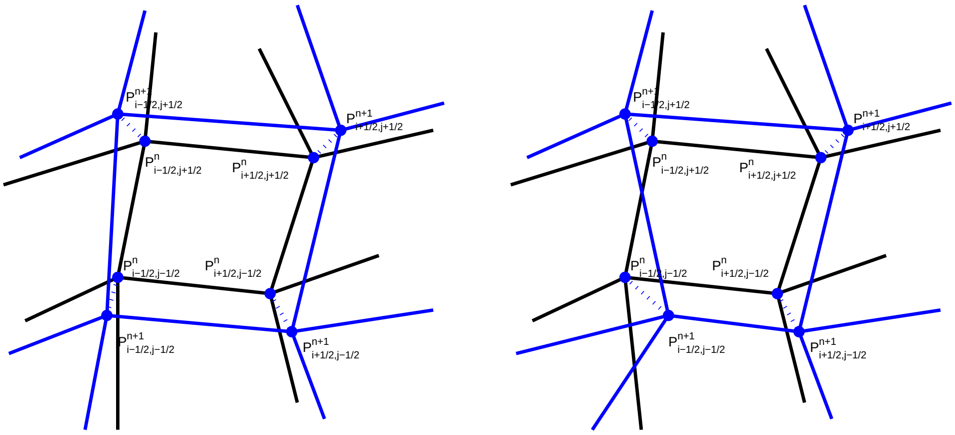

Fig. 4.2. Two situations of remapping grids. (a) Without distortion at the edge $I_{i-1/2,j}$, (b) With distortion at the edge $I_{i-1/2,j}$.

transportation of the physical quantity is not zero. So using single-edge velocity to construct numerical flux may result in large error.

At each edge $I_{i+1/2,j}$ we introduce two normal transport velocities according to the vertex velocities $\vec{u}^M(P_{i+1/2,j-1/2})$ and $\vec{u}^M(P_{i+1/2,j+1/2})$, which are

$$U_{i+\frac{1}{2},\underline{j}} = -M^{n+1/2}_{i+\frac{1}{2},j} u^{n+1/2}_{i+\frac{1}{2},j-\frac{1}{2}} + L^{n+1/2}_{i+\frac{1}{2},j} v^{n+1/2}_{i+\frac{1}{2},j-\frac{1}{2}},$$
$$U_{i+\frac{1}{2},\bar{j}} = -M^{n+1/2}_{i+\frac{1}{2},j} u^{n+1/2}_{i+\frac{1}{2},j+\frac{1}{2}} + L^{n+1/2}_{i+\frac{1}{2},j} v^{n+1/2}_{i+\frac{1}{2},j+\frac{1}{2}}.$$

Similarly at edge $I_{i,j+1/2}$, we have

$$V_{\underline{i},j+\frac{1}{2}} = B^{n+1/2}_{i,j+\frac{1}{2}} u^{n+1/2}_{i-\frac{1}{2},j+\frac{1}{2}} - A^{n+1/2}_{i,j+\frac{1}{2}} v^{n+1/2}_{i-\frac{1}{2},j+\frac{1}{2}},$$
$$V_{\bar{i},j+\frac{1}{2}} = B^{n+1/2}_{i,j+\frac{1}{2}} u^{n+1/2}_{i+\frac{1}{2},j+\frac{1}{2}} - A^{n+1/2}_{i,j+\frac{1}{2}} v^{n+1/2}_{i+\frac{1}{2},j+\frac{1}{2}}.$$

Just like the analysis to single-edge flux, the multi-point fluxes in the DCU method have the completely same form as (3.1), which are fluxes in the advection equation. The CTU scheme has fluxes

$$F^{C,M}_{i+\frac{1}{2},j} = F^{D,M}_{i+\frac{1}{2},j} - \frac{1}{2}\frac{\Delta t}{J^{n+1}_{i+1/2,j}\Delta\eta}\Big(U^{+}_{i+\frac{1}{2},\underline{j}}\mathcal{B}^{+}\Delta Q_{\bar{i},j-\frac{1}{2}} + U^{+}_{i+\frac{1}{2},\bar{j}}\mathcal{B}^{-}\Delta Q_{\bar{i},j+\frac{1}{2}}$$
$$+U^{-}_{i+\frac{1}{2},\underline{j}}\mathcal{B}^{+}\Delta Q_{\underline{i+1},j-\frac{1}{2}} + U^{-}_{i+\frac{1}{2},\bar{j}}\mathcal{B}^{-}\Delta Q_{\underline{i+1},j+\frac{1}{2}}\Big),$$

$$G^{C,M}_{i,j+\frac{1}{2}} = G^{D,M}_{i,j+\frac{1}{2}} - \frac{1}{2}\frac{\Delta t}{J^{n+1}_{i,j+1/2}\Delta\xi}\Big(V^{+}_{\underline{i},j+\frac{1}{2}}\mathcal{A}^{+}\Delta Q_{i-\frac{1}{2},\bar{j}} + V^{+}_{\bar{i},j+\frac{1}{2}}\mathcal{A}^{-}\Delta Q_{i+\frac{1}{2},\bar{j}}$$
$$+V^{-}_{\underline{i},j+\frac{1}{2}}\mathcal{A}^{+}\Delta Q_{i-\frac{1}{2},\underline{j+1}} + V^{-}_{\bar{i},j+\frac{1}{2}}\mathcal{A}^{-}\Delta Q_{i+\frac{1}{2},\underline{j+1}}\Big),$$

where $\mathcal{A}^{\pm}$ and $\mathcal{B}^{\pm}$ have the same expression as (3.4) except definitions of $U$ and $V$.

The numerical fluxes in (4.9) with second-order modification are

$$F^{C,M,2}_{i+\frac{1}{2},j} = F^{C,M}_{i+\frac{1}{2},j} + \tfrac{1}{2}(\widetilde{F}_{i+\frac{1}{2},\underline{j}} + \widetilde{F}_{i+\frac{1}{2},\bar{j}}), \quad G^{C,M,2}_{i,j+\frac{1}{2}} = G^{C,M}_{i,j+\frac{1}{2}} + \tfrac{1}{2}(\widetilde{G}_{\underline{i},j+\frac{1}{2}} + \widetilde{G}_{\bar{i},j+\frac{1}{2}}). \qquad (4.15)$$

**Remark 4.3.** Since numerical flux $F^{C,M,2}_{i+\frac{1}{2},j}$ in (4.15) is same to cells $C_{i,j}$ and $C_{i+1,j}$, the remapping scheme (4.9) is conservative.

**Remark 4.4.** Numerical scheme (4.9) with corner transport upwind and high order modification terms (4.15) satisfies discrete geometric law. In fact, when $q = 1$, these modification terms are zero. The discrete fluxes will degenerate to the DCU fluxes (4.11), which satisfies discrete GCL.

## 5. Numerical Results

In this section we will only test the numerical results of the remapping equation in the context of interpolation. That is, we will choose an underlying function, prescribe a grid motion, and compare the exact integrals of this function on the new grids with the numerical simulations. All algorithms with second-order accurate modifications use the monotinized centered limiter.

**Case 1.** The first remapped function represents a discontinuous one. It is defined as

$$q(x,y,0) = \begin{cases} 1, & if \quad -1 < x < 0, \\ 0, & if \;\; 0 < x < 1. \end{cases}$$

We carry out a simple test problem on a square domain $[-1,1] \times [-1,1]$, subdivided into a uniform grid of $20 \times 20$ cells. The special node velocity is chosen as $v = 0$ and

$$u_{i+\frac{1}{2},j+\frac{1}{2}} = \begin{cases} 1, & \text{if } \; i = 10 \text{ and } j \text{ is odd}, \\ -1, & \text{if } \; i = 10 \text{ and } j \text{ is even}, \\ 0, & \text{others}. \end{cases}$$

In this situation, the middle two-layer grid cells are stretched all the time until $t = 0.1$ when the quadrilateral meshes are degenerated to triangular ones. During this procedure, despite the overall sweep area is zero strictly, the transport of physical quantity exists. The exact integrals of this function on the new grids are 0.75 and 0.25, respectively. Using single-edge remapping method, whether the DCU method or the CTU method, the edge flux $F_{i+\frac{1}{2},j}$ is zero. The false numerical result at $t = 0.1$ is shown in Fig. 5.1. The first and second-order accurate results are the same. The multi-point flux method presented in this paper can remedy the defect. The integral on local transport area can guarantee the remapped physical quantity not being counteracted. Figs. 5.2 and 5.3 give the results using multi-point CTU method. The second-order accurate scheme is more accurate than the first-order one. The Table 5.1 compares these results on cells (10, $j$) and (11, $j$), where 'SCTU' represents single-edge CTU method and 'MCTU' is the new one proposed in this paper.

**Case 2.** The second example comes from [25]: The mesh moves according to a sequence of tensor product grids in the unit square $[0,1] \times [0,1]$, generated by

$$x^n_{i+\frac{1}{2},j+\frac{1}{2}} = x(\xi_i, \eta_j, t^n), \quad y^n_{i+\frac{1}{2},j+\frac{1}{2}} = y(\xi_i, \eta_j, t^n),$$

where $\xi_i = (i-1)/I_x, i = 1, \cdots, I_x + 1$; $\eta_j = (j-1)/J_y, j = 1, \cdots, J_y + 1$, $I_x$ and $J_y$ are mesh numbers in x-direction and y-direction, respectively, and

$$x(\xi,\eta,t) = (1-\alpha(t))\xi + \alpha(t)\xi^3, \quad y(\xi,\eta,t) = (1-\alpha(t))\eta + \alpha(t)\eta^2,$$

Table 5.1: Case 1: Comparison of classical CTU method and Multi-point flux method.

| Mesh | 1st SCTU | 2nd SCTU | 1st MCTU | 2nd MCTU | Exact |
|---|---|---|---|---|---|
| (10, j) | 1 | 1 | 0.686 | 0.786 | 0.75 |
| (11, j) | 0 | 0 | 0.314 | 0.214 | 0.25 |

Table 5.2: Case 2: The errors on the sequence tensor product grids for numerical results of 'peak' function.

| Mesh Number | $\Delta t$ | Norm | 1st MCTU | order | 2nd MCTU | order |
|---|---|---|---|---|---|---|
| 64 × 64 | 2.0e-3 | $L^1$ | 4.40e-2 | – | 3.02e-3 | – |
| 128 × 128 | 1.0e-3 | $L^1$ | 2.73e-2 | 0.69 | 1.02e-3 | 1.57 |
| 256 × 256 | 5.0e-4 | $L^1$ | 1.57e-2 | 0.80 | 3.52e-4 | 1.53 |
| 64 × 64 | 2.0e-3 | $L^\infty$ | 0.55e-0 | – | 0.13e-0 | – |
| 128 × 128 | 1.0e-3 | $L^\infty$ | 0.42e-0 | 0.39 | 8.37e-2 | 0.64 |
| 256 × 256 | 5.0e-4 | $L^\infty$ | 0.31e-0 | 0.44 | 5.82e-2 | 0.52 |

where $\alpha(t) = 0.5\sin(4\pi t)$. The final time is $t = 1$. During this time, the mesh moves along $x$-direction and $y$-direction non-uniformly. We use the anisotropic grids to check our remapping algorithm.

The remapped physical quantity is a 'peak' value function defined as

$$q(x,y,0) = \begin{cases} 0 & r > 0.25, \\ \max(0.001, 4(r-0.25)) & r \leq 0.25. \end{cases}$$

where $r = \sqrt{(x-0.25)^2 + (y-0.25)^2}$.

To discuss convergence, we first choose two norms,

$$||q - q^{ex}||_{L^\infty} = \max_{i,j} |q(x_{i,j}^N, y_{i,j}^N) - q^{ex}(x_i^N, y_j^N)|,$$
$$||q - q^{ex}||_{L^1} = \sum_{i,j} (|(q(x_{i,j}^N, y_{i,j}^N) - q^{ex}(x_{i,j}^N, y_{i,j}^N)| \cdot |C_{i,j}|),$$

where $q^{ex}$ is the exact quantity, and $q$ is the numerical results. We use three different levels of refinement to investigate convergence. The grid numbers are $I_x = J_y = 64, 128, 256$, respectively, and corresponding time steplengths decrease in proportion.

Fig. 5.4 gives the contour and 3D fully view plots for second-order multi-point flux method with 128 × 128 grid. Due to the anisotropy of the grids, the results are not perfect. The peak of

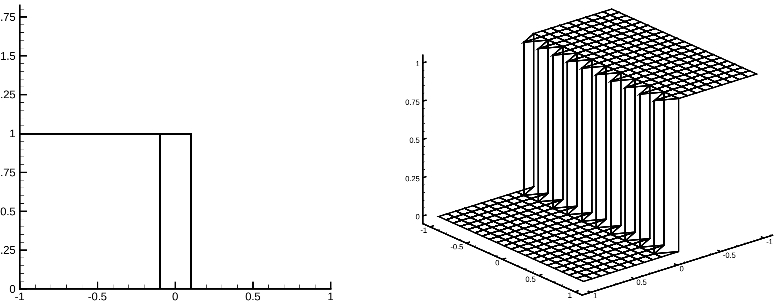

Fig. 5.1. Case 1: Classical flux methods (including DCU, CTU, the 1st-order, 2nd-order accurate) provide false results, left: 2D side view ; right: 3D full view.

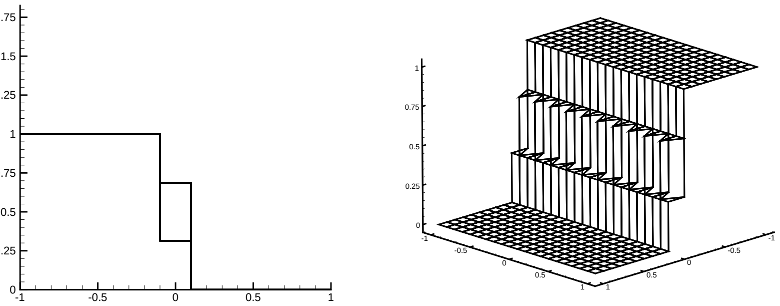

Fig. 5.2. Case 1: Multi-point flux method (first-order), left: 2D side view; right: 3D full view.

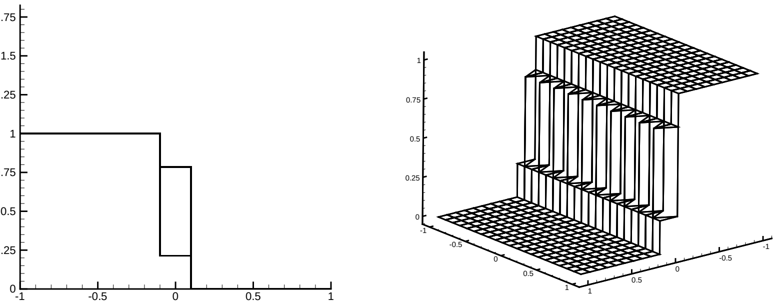

Fig. 5.3. Case 1: Multi-point flux method (second-order), left: 2D side view; right: 3D full view.

the cone is chopped off. But the method preserves the symmetry of the contour line very well. In Table 5.2 we summarize the convergence results for the sequence of tensor product grids. The $L^1$ norm convergent rates are about 1.57 and 1.53 for the two-level grids, respectively. Compared with ones in [25], these convergent rates and absolute errors are satisfying.

Obtaining such results is reasonable. In fact our scheme is monotonicity preserving due to

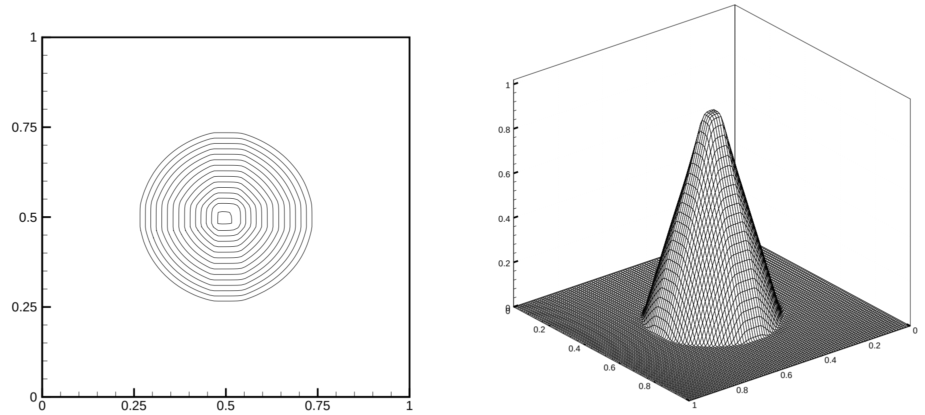

Fig. 5.4. Case 2: The ‘peak’ function on tensor product grids. Left: The contour, right: surface plot.

the use of limiter. It is well known that the high-resolution methods give rather poor accuracy at extrema (local maxima or minima in q), even though the solution is smooth. Osher and Chakravarthy [32] proved that TVD methods must in fact degenerate to first-order accuracy at extremal points. So our method is generally second-order accurate at most. With the development of TVD schemes, high-order accurate results at extremal point can be constructed see, e.g., [6, 35]. Another possibility affecting numerical results may be the anisotropy of the grids. As comparison, let us consider the next example.

**Case 3.** We will compute the same ‘peak’ value function on consecutive smoothing of an initially random grid.

Now we consider a sequence of grids, in which each is obtained from the previous one by smoothing

$$\vec{x}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}}=\frac{1}{2}\vec{x}^{n}_{i+\frac{1}{2},j+\frac{1}{2}}+\frac{\vec{x}^{n}_{i-\frac{1}{2},j+\frac{1}{2}}+\vec{x}^{n}_{i+\frac{3}{2},j+\frac{1}{2}}+\vec{x}^{n}_{i+\frac{1}{2},j-\frac{1}{2}}+\vec{x}^{n}_{i+\frac{1}{2},j+\frac{3}{2}}}{8}.$$

Table 5.3: Case 3: The errors on random smoothing grids for numerical results of ‘peak’ function.

| Mesh Number | $\Delta t$ | Norm | 1st MCTU | order | 2nd MCTU | order |
|---|---|---|---|---|---|---|
| $64 \times 64$ | 2.0e-3 | $L^1$ | 4.04e-4 | – | 1.39e-4 | – |
| $128 \times 128$ | 1.0e-3 | $L^1$ | 1.75e-4 | 1.21 | 5.55e-5 | 1.32 |
| $256 \times 256$ | 5.0e-4 | $L^1$ | 8.02e-5 | 1.13 | 2.29e-5 | 1.28 |
| $64 \times 64$ | 2.0e-3 | $L^\infty$ | 1.22e-2 | – | 6.32e-3 | – |
| $128 \times 128$ | 1.0e-3 | $L^\infty$ | 8.47e-3 | 0.53 | 5.32e-3 | 0.25 |
| $256 \times 256$ | 5.0e-4 | $L^\infty$ | 2.51e-3 | 1.75 | 1.98e-3 | 1.43 |

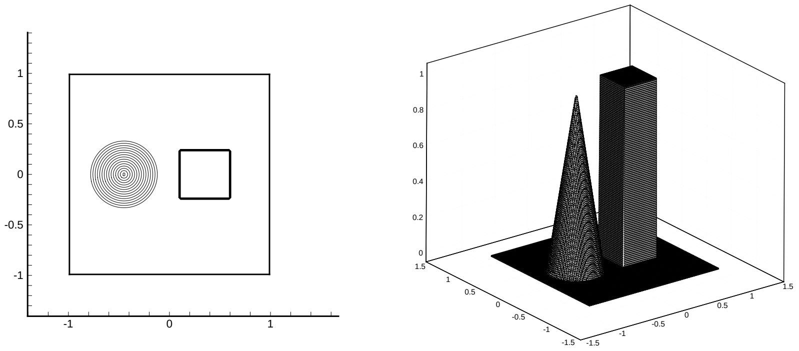

Fig. 5.5. Case 4: Initial value of a solid-body, left: contour; right: surface plots.

The initial grid is a random perturbation of a uniform grid

$$x^0_{i+\frac{1}{2},j+\frac{1}{2}} = (i-1)\Delta\xi + \gamma r_i \Delta\xi, \quad y^0_{i+\frac{1}{2},j+\frac{1}{2}} = (j-1)\Delta\eta + \gamma r_j \Delta\eta,$$

where $-0.5 \le r_i, r_j \le 0.5$ are random numbers, factor $\gamma = 0.5$, $\Delta\xi = 1/I_x$ and $\Delta\eta = 1/J_y$, $I_x$ and $J_y$ are mesh numbers in $x$ and $y$ direction, respectively. For nodes on the boundary of the unit square, only one coordinate is perturbed.

Graphically, the 'peak' value function is not distinguishable from that on tensor product grids, and so is not shown. The numerical errors and the convergent rates are presented in Table 5.3. Note that these results are more accurate than those on the sequence of tensor product grids but the convergent rates are lower. Margolin and Shashkov [25] regarded that the random sequence of grids maybe not allow a systematic buildup of remapping error, thus the results are better than those on anisotropic grids. At the same time, they regarded it should not use random grids to measure convergence rate.

**Case 4.** We will carry out a comparison of the remapping algorithm and the advection algorithm in this and next cases. The initial distribution comes from [18]. Fig. 5.5 shows an initial solid-body on a 128 × 128 grid with data $q = 0$ except in a square region where $q = 1$ and a circular region where $q$ is cone-shaped, growing to a value 1 at the center:

$$q(x,y,0) = \begin{cases} 1 & \text{if } 0.1 < x < 0.6 \text{ and } -0.25 < y < 0.25, \\ 1 - r/0.35 & \text{if } r \equiv \sqrt{(x+0.45)^2 + y^2}, \\ 0 & \text{otherwise.} \end{cases}$$

Table 5.4: Case 4: The errors in $L^1$ norm for numerical results of Case 5.

| Mesh Number | $\Delta t$ | Norm | 1st MCTU | order | 2nd MCTU | order |
|---|---|---|---|---|---|---|
| 64 × 64 | 2.0e-3 | $L^1$ | 3.86e-1 | – | 5.82e-2 | – |
| 128 × 128 | 1.0e-3 | $L^1$ | 2.88e-1 | 0.42 | 2.87e-2 | 1.02 |
| 256 × 256 | 5.0e-4 | $L^1$ | 2.04e-1 | 0.50 | 1.42e-2 | 1.02 |

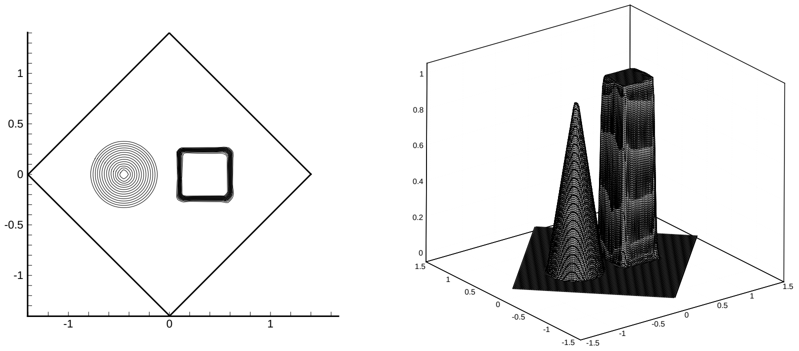

Fig. 5.6. Case 4: Time $t = \pi/8$, left: contour; right: surface plots.

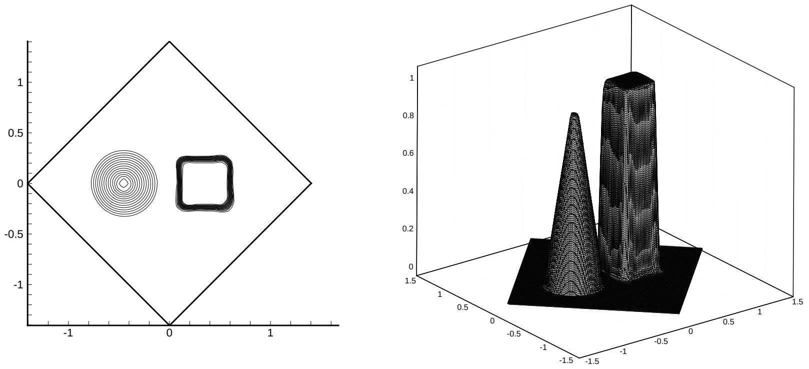

Fig. 5.7. Case 4: Time $t = 7\pi/8$, left: contour; right: surface plots.

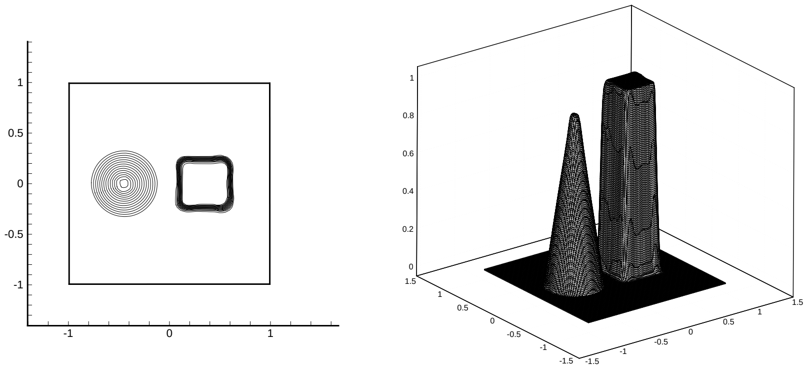

Fig. 5.8. Case 4: Time $t = \pi$, left: contour; right: surface plots.

In the remapping problem, the velocity field of the grid movement corresponds to uniform clockwise angular rotation about the center of the domain where the velocity is $u(x, y) = 2y, v(x, y) = -2x$. Obviously, at time $t = N\pi$, the grid agrees with the initial one for any integer $N$.

Figs. 5.6 – 5.8 show what we would obtain with the second-order multi-point flux scheme at time $t = \pi/8$, $t = 7\pi/8$ and $t = \pi$. The accuracy comparison at $t = \pi$ using the first and

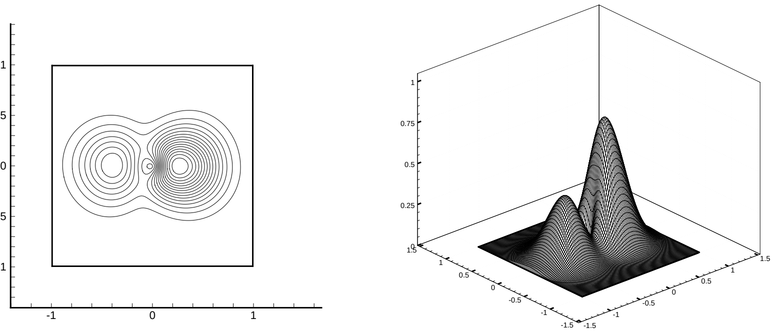

Fig. 5.9. Case 4: Time $t = \pi$, left: contour; right: surface plots. Obtained by using the first-order CTU method.

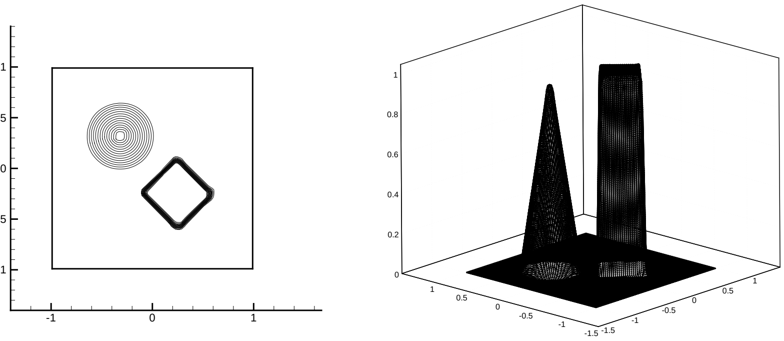

Fig. 5.10. Case 5: Advection. Time $t = \pi/8$, left: contour; right: surface plots.

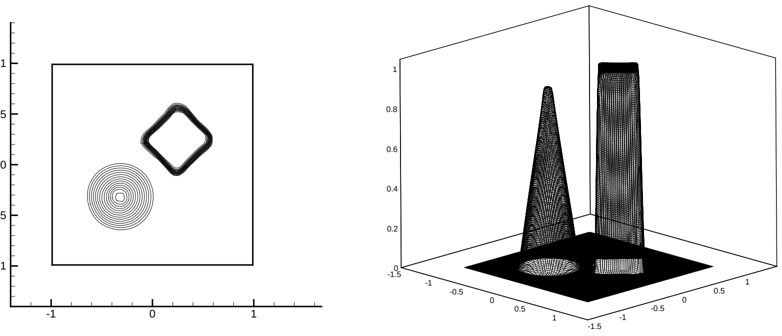

Fig. 5.11. Case 5: Advection. Time $t = 7\pi/8$, left: contour; right: surface plots.

second -order schemes are given in Table 5.4.

Once again we find the discontinuity in $q$ is smeared out, and the peak of the cone is chopped off. In particular, the convergent order just attains first for the second-order accurate scheme. However, this high-resolution method shows much better results than would be obtained with the first-order methods. Fig. 5.9 gives results with the first-order CTU method, and the

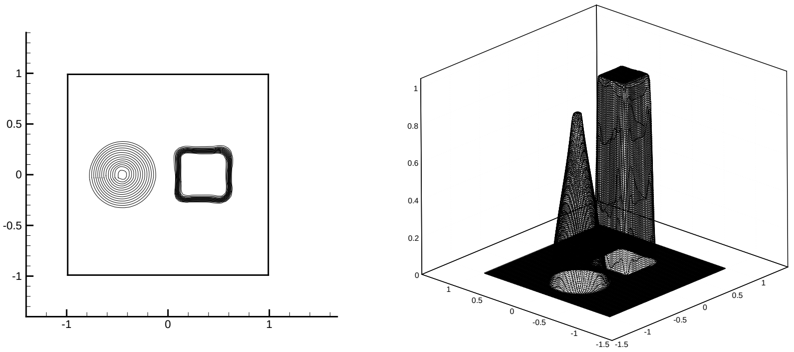

Fig. 5.12. Case 5: Advection. Time $t = \pi$, left: contour; right: surface plots.

convergent rate is just about 0.4 or so.

**Case 5.** We evaluate an advection problem corresponding to the previous remapping case. The initial value is the same as that in Case 4, and the velocity field of the solid-body $u(x, y) = 2y, v(x, y) = -2x$. Figs. 5.10 – 5.12 give the solid-body rotation results at three different times. Comparing with Figs. 5.6 – 5.8 we can see that the discontinuity resolutions between the advection and remapping algorithm are almost same.

## 6. Summary

This paper consists of two parts. One is advection algorithm; the other is the remapping one. Starting from the classical advection algorithms, we use coordinate transformation method to derive a new multi-point flux advection algorithm on an arbitrary quadrilateral grid. The new algorithm contains more information than classical CTU method. Furthermore, by introducing remapping equation, the remapping problem is shown to be equivalent to the advection one. With the help of the remapping equation, the conservative interpolation problem is transformed into solving the partial differential equation. Due to the grid motion, the remapping algorithm is not completely equivalent to the advection one, and more conditions need to be satisfied, such as discrete geometric conservation law. In addition, to obtain overall second-order accurate scheme in form, the remapping algorithm needs to discretize more terms in the associated truncation error expansion than the advection one. But if neglecting these extra terms, the flux formulations in the remapping algorithm are almost the same as those in advection scheme (see Remark 4.2). Once more we use coordinate transformation method to construct a new multi-point remapping algorithm suitable for continuous rezone ALE procedures. It can be regarded as a natural generalization of multi-point flux advection algorithm.

The new multi-point flux method retains many advantageous properties, such as conservation, accuracy, and including corner-coupling terms. Notable among these is the consistency with node local motion, which can remedy partial defects of the conventional CTU method. A numerical experiment validates this point. The other numerical tests either show that the scheme has good robustness on anisotropic grids, or illustrate the accuracy on random grids. In general, the new remapping method has almost the same resolution as corresponding advection algorithm.

Essentially, the process of deriving remapping schemes in this paper provides a framework to transform an algorithm on a rectangular grid into the one on moving grid. In this framework, we may construct algorithm not only for linear remapping equation, but also for nonlinear scalar equation and nonlinear systems of conservation law. The next work is to construct a moving mesh method for gas hydrodynamics equations. The feature of the new algorithm will include corner coupling information and be consistent with grid motion. Of course, due to the nonlinearity and its viscosity properties, the finite volume method for system of conservation law may suffer from numerical shock instability or other nonphysical phenomena (see, e.g., [34,36]), more modification techniques need to be adopted.

**Acknowledgments.** This project was supported by the National Natural Science Foundation of China (11071025), the Foundation of CAEP (2010A0202010), the Foundation of Laboratory of Science and Technology on Computational Physics and the Defense Industrial Technology Development Program (B1520110011).

## References

[1] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, *Comput. Methods Appl. Mech. Engrg.*, **99** (1992), 235-394.

[2] D.J. Benson, Momentum advection on a staggered mesh, *J. Comput. Phys.*, **100** (1992), 143-162.

[3] J. Cheng and C.W. Shu, A high order accurate conservative remapping method on staggered meshes, *Appl. Numer. Math.*, **58**:7 (2008), 1042-1060.

[4] J. Cheng, C.W. Shu and Q.H. Zeng, A conservative Lagrangian scheme for solving compressible fluid flows with multiple internal energy equations, *Commun. Comput. Phys.*, **12** (2012), 1307-1328.

[5] P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, *J. Comput. Phys.*, **87** (1990), 171-200.

[6] P. Colella, M.D. Sekora, A limiter for PPM that preserves accuracy at smooth extrema, *J. Comput. Phys.*, **227** (2008), 7069-7076.

[7] B. Després and C. Mazeran, Lagrangian gas dynamics in two dimensions and Lagrangian systems, *Arch. Rational Mech. Anal.*, **178** (2005), 327-372.

[8] X. G. Deng, M. L. Mao, G. H. Tu, H. X. Zhang and Y. F. Zhang, High-order and high accurate CFD methods and their applications for complex grid problems, *Commun. Comput. Phys.*, **11** (2012), 1081-1102.

[9] J. Dukowicz and J. Baumgardner, Incremental remapping as a transportation/advection algorithm, *J. Comput. Phys.*, **160** (2000), 318-335.

[10] C. Farhat, P. Geuzaine and C. Grandmont, The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids, *J. Comput. Phys.*, **174**:2 (2001), 669-694.

[11] S. Galera, P. H. Maire and J. Breil, A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction, *J. Comput. Phys.*, **229** (2010), 5755-5787.

[12] J. Grandy, Conservative remapping and regions overlays by intersecting arbitrary polyhedra, *J. Comput. Phys.*, **148** (1999), 433-466.

[13] H. Guillard and C. Farhat, On the signicance of the geometric conservation law for flow computations on moving meshes, *Comput. Meth. Appl. Mech. Engrg.*, **190**:11-12 (2000), 1467-1482.

[14] P. Hoch, S. Marchal, Y. Vasilenko and A. A. Feiz, Nonconformal adaptation and mesh smoothing for compressible Lagrangian fluid dynamics, ESAIM: PROC, **24** (2008), 111-129.

[15] M. Kucharik, M. Shashkov and B. Wendroff, An efficient linearity-and-bound-preserving remapping method, *J. Comput. Phys.*, **188** (2003), 462-471.

[16] B. van Leer, Multidimensional Explicit Difference Schemes for Hyperbolic Conservation Laws, Computing Methods in Applied Sciences and Engineering, VI, Elsevier Science, New York, 1984.

[17] M. Lesoinne and C. Farhat, Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact of aeroelastic computations, *Comput. Methods Appl. Mech. Engrg.*, **134** (1996), 71-90.

[18] R. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge Univeristy Press, Cambridge, 2002.

[19] R. Leveque and R. Walder, Grid alignment effects and rotated methods for computing complex flows in astrophysics, GAMM Conf. on Comput. Fluid Dyn. Lausanne, 1991.

[20] R. Loubere, P. H. Maire, M. Shashkov, J. Breil and S. Galera, ReALE: A reconnection-based arbitrary-LagrangianCEulerian method, *J. Comput. Phys.*, **229** (2010), 4724-4761.

[21] P.H. Maire, A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes, *J. Comput. Phys.*, **228** (2009), 2391–2425.

[22] P.H. Maire, A high-order cell-centered Lagrangian scheme for compressible fluid flows in two-dimensional cylindrical geometry, *J. Comput. Phys.*, **228** (2009), 6882-6915.

[23] P. H. Maire, R. Abgrall, J. Breil and J. Ovadia, A cell-centered Lagrangian scheme for compressible flow problems, *SIAM J. Sci. Comput.*, **29**:4 (2007), 1781-1824.

[24] L. G. Margolin and C. W. Beason, Remapping on the staggered mesh, Report UCRL-99682 of Lawrence Livermore National Laboratory, 1988.

[25] L. Margolin and M. Shashkov, Second-order sign-preserving remapping on general grids, *J. Comput. Phys.*, **184** (2003), 266-298.

[26] L. Margolin and P. Smolarkiewicz, Antidiffusive velocities for multipass donor cell advection, *SIAM J. Sci. Comput.*, **20** (1998), 907-929.

[27] D. J. Mavriplis and Z. Yang, Construction of the discrete geometric conservation law for high-order time-accurate simulations on dynamic meshes, *J. Comput. Phys.*, **213**:2 (2006), 557-573.

[28] S. Mosso, D. Burton, et al., A second order, two- and three-dimensional remap method, LA-UR-98-5353, Report of Los Alamos National Laboratory, Los Alamos, NM, USA, 1988.

[29] S. Mosso and B. Swartz, An unsplit, two-dimensional advection algorithm, LA-UR-01-1476, Report of Los Alamos National Laboratory, Los Alamos, NM, USA, 2000.

[30] P. J. O'Rourke and M. S. Sahota, A variable explicit/implicit numerical method for calculating advection on unstructured meshes, *J. Comput. Phys.*, **143** (1998), 312-345.

[31] A. L. Ortega and G. Scovazzi, A geometrically-conservative, synchronized, flux-corrected remap for arbitrary Lagrangian-Eulerian computations with nodal finite elements, *J. Comput. Phys.*, **230** (2011), 6709-6741.

[32] S. Osher and S. Chakravarthy, High resolution schemes and the entropy condition, *SIAM J. Numer. Anal.*, **21** (1984), 955-984.

[33] R. Pember and R. Anderson, Comparison of direct Eulerian Godunov and Lagrange plus remap articial viscosity schemes for compressible flow, Technical Report AIAA Paper 2001-2644, 2001.

[34] J. Quirk, A contribution to the great Riemann solver debate, *Int. J. Numer. Meth. Fluids*, **18** (1994), 555-574.

[35] W.J. Rider, J.A. Greenough and J.R. Kamm, Accurate monotonicity- and extrema- preserving methods through adaptive nonlinear hybridizations, *J. Comput. Phys.*, **225** (2007), 1827-1848.

[36] Z.J. Shen, W. Yan and G.X. Lv, Behavior of viscous solutions in Lagrangian formulation, *J. Comput. Phys.*, **229** (2010), 4522-4543.

[37] R. Smith, AUSM(ALE): A geometrically conservative arbitrary Lagrangian-Eulerian flux splitting scheme, *J. Comput. Phys.*, **150** (1999), 268-286.

[38] P.D. Thomas and C.K. Lombard, Geometric conservation law and its application to flow computations on a moving grid, *AIAA J.*, **17**:10 (1979), 1030-1037.

[39] F. Vilar and P.H. Maire and R. Abgrall, Cell-centered discontinuous Galerkin discretizations for two-dimensional scalar conservation laws on unstructured grids and for one-dimensional La-

grangian hydrodynamics, *Comput. Fluids*, **46** (2011), 498-504.

[40] L. White and A. Adcroft, A high-order finite volume remapping scheme for nonuniform grids: the piecewise quartic method (PQM), *J. Comput. Phys.*, **277** (2008), 7394-7422.