ACCELERATED PISTON PROBLEM AND HIGH ORDER MOVING BOUNDARY TRACKING METHOD FOR COMPRESSIBLE FLUID FLOWS[*]

ZHIFANG DU[†] AND JIEQUAN LI[‡]

Abstract. Reliable tracking of moving boundaries is important for the simulation of compressible fluid flows, and there are a lot of contributions in literature. We recognize from the classical piston problem, a typical moving boundary problem in gas dynamics, that acceleration is a key element in the description of the motion and should be incorporated into the design of a moving boundary tracking method. Technically, the resolution of the accelerated piston problem boils down to a one-sided generalized Riemann problem solver, which is taken as the building block for constructing schemes with high order accuracy in both space and time. In this paper we take this into account, together with the cell-merging approach, to propose a new family of high order accurate moving boundary tracking methods and verify its performance through one- and two-dimensional test problems, along with accuracy analysis.

Key words. compressible fluid flows, accelerated piston problem, moving boundary tracking method, one-sided generalized Riemann problem solver, cell-merging criterion

AMS subject classifications. 35L50, 35L65, 65M08, 76M12, 76N15

DOI. 10.1137/19M1266599

1. Introduction. Moving boundary problems are ubiquitous for engineering applications; in particular, the tracking of moving boundaries is an essential technique determining the quality of underlying simulations for compressible fluid flows. There are many studies in this context, e.g., the front tracking method [19], the moving boundary tracking (MBT) method [14, 35], the immersed boundary method [30], level set methods [17], volume of fluid methods [18], moment of fluid methods [13], and adaptive mesh refinement methods [7], to name just a few. In this paper, we will follow the MBT method in the finite volume framework, with the new recognition of acceleration as a key role in the description of moving boundaries, to develop a high order accurate version. The study is motivated by a basic moving boundary problem: the accelerated piston problem in gas dynamics [8].

Assume that a uniform piston with a thickness L moves in a tube filled with gas. Its motion is described by two elements, the velocity and the acceleration [36],

$$
(1) \quad
\begin{aligned}
\frac{dx_c(t)}{dt} &= u_c(t), \\
\frac{du_c(t)}{dt} &= -\mathcal{S}_c \frac{p(x_c(t) + L/2 + 0, t) - p(x_c(t) - L/2 - 0, t)}{\mathcal{M}_c},
\end{aligned}
$$

†Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Beijing, 100088, People's Republic of China (du@mail.bnu.edu.cn).
‡Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Beijing, 100088, People's Republic of China, and Center for Applied Physics and Computational Mathematics, Peking University, Beijing, 100871, People's Republic of China (li_jiequan@iapcm.ac.cn).

where $x_c(t)$, $u_c(t)$, and $\mathcal{M}_c$ are the location, the velocity, and the mass of the piston, $\mathcal{S}_c$ is the sectional area of the tube, and $p(x_c(t) \pm L/2 \pm 0, t)$ are pressures exerted on the two faces of the piston. The two equations in (1) represent the Newtonian first and second laws for fluid flows, respectively, and they are coupled with the Euler equations, leading to a coupled dynamical system [23]. This might be the simplest moving (free) boundary problem in the context of compressible flows. It is natural to use the pair $(u_c(t), du_c(t)/dt)$ for the tracking of this piston, more or less like the symplectic algorithm [15],

$$(2) \qquad x_c(t_{n+1}) = x_c(t_n) + \Delta t u_c(t_n) + \frac{\Delta t^2}{2} \frac{du_c(t_n)}{dt} + \mathcal{O}(\Delta t^3),$$

where $\Delta t = t_{n+1} - t_n$ is the time increment. Otherwise, one would have to use Runge–Kutta-type time stepping to advance the trajectory of the piston, for which only the velocity $u_c(t)$ is adopted. As the piston is resolved together with the evolution of the fluid flow in the whole flow region, the problem boils down to the one-sided generalized Riemann problem (GRP), the so-called initial boundary value problem (IBVP) with the piston as a free boundary. Numerically, when the finite volume framework is adopted, it is necessary to develop a one-sided GRP solver in order to construct numerical fluxes on the piston surface and track the moving boundary (piston). The present study just establishes the interrelation between the resolution of the accelerated piston problem and the MBT method, leading to a high order version of MBT.

The MBT method originated in [14], and its advantage was clearly stated there. Since the uniformity of flow variables is assumed in boundary cells and the one-sided Riemann solver is adopted to compute the corresponding numerical fluxes, the MBT algorithm developed there has at most first order accuracy in boundary cells [14, p. 91]. The present contribution uses the direct Eulerian GRP solver [6], which enables us to develop a one-sided GRP solver suited for the MBT methodology to achieve high order accuracy. The boundary is tracked with high order accuracy simultaneously, and no extra technology is needed. In addition, since the present GRP solver has been updated for multidimensional computations [26], the dimensional operator-splitting algorithm in [14] is not necessary. As far as complex geometries are concerned, the present paper adopts the cell-merging approach in [34, 33] to deal with the "small-cell" problem. Small cut cells are absorbed by their neighbors to form larger control volumes, and fluid states are evolved over time-varying control volumes so that the resulting MBT scheme is consistent with the integrated form of conservation laws. This is different from the body-fitted or adaptive grid schemes [2, 28, 10, 32]. The MBT is also applied in other frameworks, e.g., the wave propagation algorithm [35]. We particularly refer to the inverse Lax–Wendroff (ILW) method [37], which interpolates the ghost fluid state with high order spatial accuracy by taking the boundary acceleration into consideration. The temporal accuracy is obtained by the popular Runge–Kutta discretization.

This paper is organized as follows. Basic facts about rigid body motion and the MBT method are given in section 2. One- and two-dimensional one-sided GRP solvers are developed in section 3. The MBT scheme is proposed in section 4. The numerical accuracy of the newly developed MBT method is analyzed in section 5. Numerical experiments involving fluid-surface interactions are carried out in section 6 for the performance. Some discussions are made in section 7.

2. Accelerated piston problem and moving boundary tracking method. This section describes the piston problem, which motivates the new high order MBT
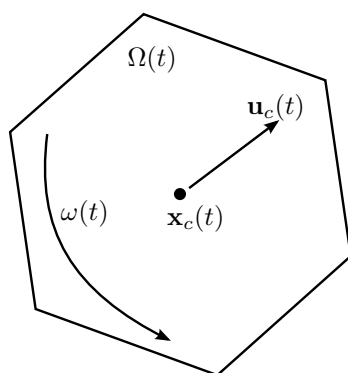
FIG. 1. *The motion of a rigid body $\Omega(t)$ in the compressible fluid is described with its barycenter $\mathbf{x}_c(t)$, a translational $\mathbf{u}_c(t)$, and a rotational velocity $\omega(t)$.*

scheme that we are going to propose. The key observation is that the acceleration of a boundary should be taken as a natural and necessary element to describe its motion. Let us describe the rigid body motion in one and two space dimensions.

2.1. Rigid body motion and accelerated piston problem. Think of a rigid body, represented by the polygon $\Omega(t)$ with its barycenter locating at $\mathbf{x}_c(t)$, moving in the compressible fluid with a translational velocity $\mathbf{u}_c(t)$ and a rotational velocity $\omega(t)$, as shown in Figure 1. The rotational velocity ω is regarded as a pseudovector, which is positive if Ω rotates counterclockwise. The flow field around this body is described by the compressible Euler equations (only two-dimensional in the present paper, but similarly for three-dimensional cases),

$$
\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{W})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{W})}{\partial y} = 0,
$$

(3)
$$
\mathbf{W} = (\rho, \rho u, \rho v, \rho E)^\top,
$$

$$
\mathbf{F}(\mathbf{W}) = (\rho u, \rho u^2 + p, \rho uv, u(\rho E + p))^\top,
$$

$$
\mathbf{G}(\mathbf{W}) = (\rho v, \rho uv, \rho v^2 + p, v(\rho E + p))^\top,
$$

where $\mathbf{u} = (u, v)$ is the velocity in the $\mathbf{x} = (x, y)$-coordinates, ρ is the density, p is the pressure, and $E = \frac{1}{2}(u^2 + v^2) + e$ is the total energy. The internal energy e is determined by the equation of state (EOS) $e = e(\rho, p)$. Thermodynamical quantities satisfy the Gibbs relation,

(4)
$$
Tds = de + pd\tau,
$$

where $\tau = 1/\rho$ and s is the entropy. For polytropic gases, $p = (\gamma - 1)\rho e$ and $\gamma > 1$ is the specific heat ratio.

The boundary of the rigid body $\Omega(t)$ is denoted by $\partial\Omega(t)$ with the unit normal vector $\mathbf{n}$ pointing from the fluid to the rigid body. Then the motion of the body obeys

(5)
$$
\frac{d\mathbf{x}_c(t)}{dt} = \mathbf{u}_c(t), \quad \frac{d\mathbf{u}_c(t)}{dt} = \frac{1}{\mathcal{M}_c} \int_{\partial\Omega(t)} p\, \mathbf{n}\, dl,
$$

and its rotational motion is described using the equations

(6)
$$
\frac{d\theta}{dt} = \omega(t), \quad \frac{d\omega}{dt} = \frac{1}{\mathcal{A}_c} \int_{\partial\Omega(t)} p\, (\mathbf{x} - \mathbf{x}_c(t)) \times \mathbf{n}\, dl,
$$

where $\mathcal{M}_c$ is the mass of the solid body, $\theta(t)$ and $\omega(t)$ are the relative angle and the angular velocity of $\Omega(t)$ with respect to its barycenter $\mathbf{x}_c(t)$, and $\mathcal{A}_c$ is the inertia of the solid body. For all $\mathbf{X}_b \in \partial\Omega(0)$, denote by $\mathbf{x}_b(t; \mathbf{X}_b)$ its trajectory from $\mathbf{X}_b$ for $t > 0$. The instantaneous translational motion of any point $\mathbf{x}_b(t; \mathbf{X}_b) \in \partial\Omega(t)$ is described by

$$(7) \qquad \begin{aligned} \frac{d\mathbf{x}_b(t; \mathbf{X}_b)}{dt} &= \mathbf{u}_c(t) + \omega(t)\, \mathbf{r}_b^{\perp}(t; \mathbf{X}_b), \quad \mathbf{x}_b(0; \mathbf{X}_b) = \mathbf{X}_b, \\ \frac{d\mathbf{u}_b(t; \mathbf{X}_b)}{dt} &= \frac{d\mathbf{u}_c}{dt} + \frac{d\omega(t)}{dt}\, \mathbf{r}_b^{\perp}(t; \mathbf{X}_b) + \omega(t)\Big[\big(\omega(t)\, \mathbf{r}_b^{\perp}(t; \mathbf{X}_b)\big)\Big]^{\perp}, \end{aligned}$$

where $\mathbf{r}_b(t; \mathbf{X}_b) = \mathbf{x}_b(t; \mathbf{X}_b) - \mathbf{x}_c(t)$, $\mathbf{u}_c(0)$ and $\omega(0)$ are prescribed, and $\mathbf{u}_b$ is the velocity of the boundary point $\mathbf{x}_b(t; \mathbf{X}_b) \in \partial\Omega(t)$. We use the notation $\mathbf{a}^{\perp} = (-a_y, a_x)^{\top}$ for any vector $\mathbf{a} = (a_x, a_y)^{\top}$. Assume for the time being that the motion is not affected by surroundings in order to avoid the complications caused by the interaction with other boundaries. In order to describe the interaction between the rigid body and the compressible fluid, a proper boundary condition along the moving boundary $\partial\Omega(t)$ is required,

$$(8) \qquad \mathbf{u}(\mathbf{x}_b(t; \mathbf{X}_b), t) \cdot \mathbf{n}(t; \mathbf{X}_b) = \mathbf{u}_b(t; \mathbf{X}_b) \cdot \mathbf{n}(t; \mathbf{X}_b) \quad \forall \mathbf{x}_b(t; \mathbf{X}_b) \in \partial\Omega(t),$$

where $\mathbf{n}(t; \mathbf{X}_b)$ is the unit normal vector of $\partial\Omega(t)$ at $\mathbf{x}_b(t; \mathbf{X}_b)$, pointing from the fluid into the rigid body.

When the problem reduces to one-dimensional cases, the rigid body motion in the flow field corresponds to the *free piston problem* [8, 36], as described in the introduction. As far as the motion of the piston is concerned, we ignore the discussion on both ends of the cylinder and assume that this piston moves over the whole line. The thickness L can be also assumed to be infinitesimal. Then $p(x_c(t) + 0, t)$ and $p(x_c(t) - 0, t)$ are obtained by simultaneously solving the following two (left and right) free boundary problems:

$$(9) \qquad \begin{aligned} \frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{W})}{\partial x} &= 0, \quad x_c(t) < x < \infty, \quad t > 0, \\ \mathbf{W}(x, 0) &= \mathbf{W}_+(x), \quad x_c(0) < x < \infty, \\[2mm] u(x_c(t), t) &= u_c(t), \quad x_c(t) = x_c(0) + \int_0^t u_c(s)\,ds, \end{aligned}$$

and

$$(10) \qquad \begin{aligned} \frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{W})}{\partial x} &= 0, \quad -\infty < x < x_c(t), \quad t > 0, \\ \mathbf{W}(x, 0) &= \mathbf{W}_-(x), \quad -\infty < x < x_c(0). \\[2mm] u(x_c(t), t) &= u_c(t), \quad x_c(t) = x_c(0) + \int_0^t u_c(s)\,ds, \end{aligned}$$

where the velocity $u_c(t)$ is defined through (1). Note that the pressure gradient determines the acceleration, and the pressure exerted on the piston is not identical in general. Therefore, $u_c(t)$ is not constant in t and the piston is accelerated. This is called the *accelerated piston problem*. This observation will be put in the design of the high order MBT method in this paper.

2.2. Moving boundary tracking methods. The high order moving boundary tracking (MBT) method we are going to propose works in the finite volume framework, along with the cell-merging algorithm. Since the control volumes are Eulerian in the interior of the computational domain, we just focus on boundary control volumes. For easy understanding of the presentation, we first describe the one-dimensional version and then the two-dimensional case.

In one dimension, we consider the piston problem from the left and assume that the fluid occupied region is $\bigcup_{0<t<T}[-\infty, x_c(t)]$ by ignoring the influence of the far field in the left. The computational domain is divided into uniform cells,

$$(11) \qquad \mathcal{T} = \bigcup_j \left\{ I_j = (x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}) : x_{j+\frac{1}{2}} = \left(j + \frac{1}{2}\right) \Delta x, \; j \in \mathbb{Z} \right\}.$$

We focus on the moving boundary (piston) cell. Assume that the piston is located in the rightmost cell I_J indexed by J, i.e., $x_c(t_n) \in I_J$. Denote by $I_J^c(t) = (x_{J-\frac{1}{2}}, x_c(t))$ the "legal" cell according to the following cell-merging criterion. Then we apply (3) over the space-time control volume $B_J(t_n, t_{n+1}) := \{(x,t); x \in I_J^c(t), t_n \leq t \leq t_{n+1}\}$,

$$(12)$$
$$\int_{I_J^c(t_{n+1})} \mathbf{W}(x, t_{n+1}) dx = \int_{I_J^c(t_n)} \mathbf{W}(x, t_n) dx$$
$$- \left\{ \int_{t_n}^{t_{n+1}} \left[\mathbf{F}(\mathbf{W}(x_c(t) - 0, t)) - u_c(t) \mathbf{W}(x_c(t) - 0, t) \right] dt \right.$$
$$\left. - \int_{t_n}^{t_{n+1}} \mathbf{F}(\mathbf{W}(x_{J-\frac{1}{2}}, t)) dt \right\},$$

where $\mathbf{W} = (\rho, \rho u, \rho E)^\top$ and $\mathbf{F}(\mathbf{W}) = (\rho u, \rho u^2 + p, u(\rho E + p))^\top$. The MBT method consists of three ingredients: *the flow evolution, the boundary tracking, and the cell merging.* As usually implemented for the flow evolution, we need to approximate the fluxes properly. The interior flux $\int_{t_n}^{t_{n+1}} \mathbf{F}(\mathbf{W}(x_{J-\frac{1}{2}}, t)) dt$ is evaluated using the standard GRP solver [6]. However, the boundary flux depends on the one-sided GRP solver that provides the values

$$(13) \; \mathbf{W}_{c,-}^{n,*} := \lim_{t \to t_n+0} \mathbf{W}(x_c(t) - 0, t), \quad \left(\frac{d\mathbf{W}}{dt}\right)_{c,-}^{n,*} := \lim_{t \to t_n+0} \left(\frac{d_c \mathbf{W}}{dt}\right)(x_c(t) - 0, t),$$

where $d_c/dt = \partial/\partial t + u_c(t)\partial/\partial x$ is the directional derivative along the boundary $x = x_c(t)$. This pair of values also serves to track the boundary, as expressed in (2).

As the piston travels to the next time level $t = t_{n+1}$, the boundary cell $(x_{J-\frac{1}{2}}, x_c(t_{n+1}))$ could be very small or large so that it should be redistributed, obeying the cell-merging criterion described below.

One-dimensional cell-merging criterion.

   a. If $x_c(t_{n+1}) \in I_J$ and $|I_J^c(t_{n+1})| > \kappa \Delta x$, where κ is a user-tuned parameter, then $I_J^c(t_{n+1})$ is still well-defined. Otherwise, I_{J-1} and $I_J^c(t_{n+1})$ are merged to form a new boundary cell, denoted $I_{J-1}^c(t_{n+1})$.

   b. If $x_c(t_{n+1}) \in I_{J+1}$, the boundary control volume is $I_J^c(t) = (x_{J-\frac{1}{2}}, x_c(t_{n+1}))$.

   c. If $x_c(t_{n+1}) \in I_{J-1}$, the boundary cell is $I_{J-1}^c(t_{n+1}) = (x_{J-\frac{3}{2}}, x_c(t_{n+1}))$.

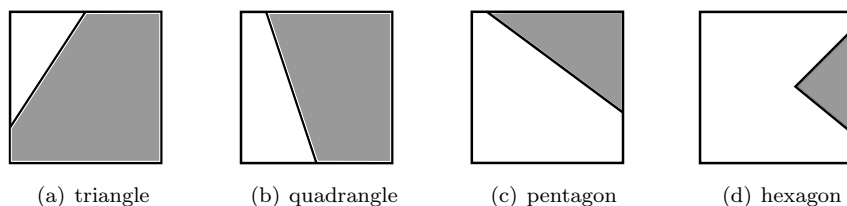(a) triangle   (b) quadrangle   (c) pentagon   (d) hexagon

FIG. 2. *Illustration of cut cells. A cut cell may be either a triangle, a quadrangle, a pentagon, or a hexagon: The shaded parts represent the solid body $\Omega(t)$.*

In two dimensions, we still use the Cartesian meshes,

$$
\mathcal{T} = \bigcup_{jk} \{ I_{jk} = (x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}) \times (y_{k-\frac{1}{2}}, y_{k+\frac{1}{2}}):
$$
(14)
$$
x_{j+\frac{1}{2}} = (j+\tfrac{1}{2})\Delta x, \ y_{k+\frac{1}{2}} = (k+\tfrac{1}{2})\Delta y, \ j,k \in \mathbb{Z} \}.
$$

The Cartesian computational cells are usually cut by the boundary $\partial\Omega$ of solid objects to form very small "illegal" cells. We use the cell-merging approach to modify boundary cells [33]. In general, the shape of a cut cell may be a triangle, a quadrangle, a pentagon, or a hexagon, as shown in Figures 2(a)–2(d). In the first three cases, the presence of any node of $\partial\Omega$ inside the cut cell makes no difference in terms of the cell-merging procedure stated below since what matters is the topological shape of the cut cell. The last hexagon case represents the situation that the corner of approximated $\partial\Omega$ is located inside a cut cell and intersects a straight segment, as shown in Figure 2(d). The current study assumes the corners on $\partial\Omega$ are broad enough to avoid more complex cases.

For the time being, we assume I_{JK}^c is a "legal" boundary cell surrounded by the interior interface $\Gamma_{int}(t)$ and the moving boundary $\Gamma_b(t)$. Denote by $B_{JK}(t_n, t_{n+1}) = \{(x,y,t); (x,y) \in I_{JK}^c(t), t_n \le t \le t_{n+1}\}$ the space-time control volume. As the flow equations (3) are applied over this control volume, we obtain
(15)

$$
\int_{I_{JK}^c(t_{n+1})} \mathbf{W}(\mathbf{x}, t_{n+1}) d\mathbf{x}
$$
$$
= \int_{I_{JK}^c(t_n)} \mathbf{W}(\mathbf{x}, t_n) d\mathbf{x}
$$
$$
- \int_{t_n}^{t_{n+1}} \left\{ \int_{\Gamma_{int}(t)} (\mathbf{F}, \mathbf{G})^\top \cdot \mathbf{n} \, dl dt + \int_{\Gamma_b(t)} \left[(\mathbf{F}, \mathbf{G})^\top \cdot \mathbf{n} - (\mathbf{u} \cdot \mathbf{n})\mathbf{W} \right] dl dt \right\},
$$

where $\mathbf{n}$ is the unit outer normal vector of $\partial I_{JK}^c(t)$. In parallel to the one-dimensional case, we develop a two-dimensional one-sided GRP solver to provide the instantaneous values

$$
(16) \qquad \mathbf{W}_b^{n,*} = \lim_{t \to t_n + 0} \mathbf{W}(\mathbf{x}_b(t), t), \qquad \left(\frac{d\mathbf{W}}{dt} \right)_b^{n,*} = \lim_{t \to t_n + 0} \frac{d_c\mathbf{W}}{dt}(\mathbf{x}_b(t), t),
$$

where $\mathbf{x}_b(t_n)$ is any point on Γ_b at $t = t_n$, and $d_c/dt = \partial/\partial t + u_c \partial/\partial x + v_c \partial/\partial y$. These values serve to approximate the flux and track the moving boundary in the spirit of the GRP method.

The tracking of the moving boundary Γ_b is much more involved, compared to the one-dimensional counterpart, and is described by (7). For the efficiency of the

algorithm, the cell-merging procedure is still necessary to avoid very small boundary cells.

The cell-merging procedure is stated briefly as follows. Suppose that the volume of $I_{JK}^c(t)$ is small enough, i.e., $|I_{JK}^c(t)| < \kappa \Delta x \Delta y$, where κ is a user-tuned parameter. Then it should be absorbed by its neighbors to form a larger "legal" computational cell. Denote the normal vector at the segment $\partial \Omega(t) \cup I_{JK}$ by $\mathbf{n}_{JK}(t) = (n_{JK}^x(t), n_{JK}^y(t))^\top$.

Two-dimensional cell-merging criterion.
a. If $I_{JK}^c(t_n)$ is a triangle, combine it with its neighbor. For example, consider the situation shown in Figure 3(a), where $|n_{JK}^x(t_n)| > |n_{JK}^y(t_n)|$, and combine I_{JK} with $I_{J-1,K}$. Otherwise, combine I_{JK} with $I_{J,K+1}$.
b. If $I_{JK}^c(t_n)$ is a quadrangle, combine it with its uncut neighbor. For example, consider the situation shown in Figure 3(b), and combine I_{JK} with $I_{J-1,K}$.
c. Consider the situation that the cut cells are pentagons. Assume that a pentagon has two cut edges. Then there are two cases for the merging, depending on their lengths.
   (i) If its horizontal (resp., vertical) cut edge is shorter than $\kappa \Delta x$ (resp., $\kappa \Delta y$), combine the pentagon with its neighbor to lengthen this cut edge, as shown in Figure 3(c). In this paper κ is taken as 0.3 for all numerical tests.
   (ii) Otherwise, leave the pentagon as a "legal" cell.
d. If $I_{JK}^c(t_n)$ is a hexagon, combine it with its multiple-cut neighbor to form a "big" cell, as shown in Figure 3(d).

(a) The cell merging of a triangle

(b) The cell merging of a quadrangle

(c) The cell merging of a pentagon

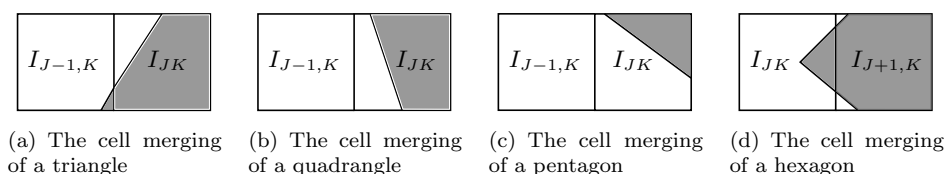(d) The cell merging of a hexagon

FIG. 3. *The cell-merging procedure of cut cells.*

3. One-sided generalized Riemann problem solver. In order to develop a high order MBT scheme in the finite volume framework, we need to approximate the flux with a high order accuracy, which boils down to solving the following one-sided generalized Riemann problem (GRP). In this sense, the accelerated piston problem (9) is also called the *one-sided generalized Riemann problem.* Two cases are discussed: the one-dimensional one-sided GRP solver and the two-dimensional one-sided GRP solver. The difference between them is that the transversal effect relative to the boundary interface is incorporated into the two-dimensional GRP solver.

3.1. One-sided generalized Riemann problem solver in one dimension. The one-sided GRP for the Euler equations in one space dimension is formulated as

$$(17) \quad \begin{aligned} &\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{W})}{\partial x} = 0, && -\infty < x < x_c(t), \ t > 0, \\ &\mathbf{W}(x,0) = \mathbf{W}_-(x), && -\infty < x < 0, \\ &u(x_c(t), t) = u_c(t), && x_c(t) = x_c(0) + \int_0^t u_c(s) ds, \end{aligned}$$

where $\mathbf{W}_-(x)$ is a smooth vector function and usually taken as a polynomial for numerical purposes. The boundary trajectory $x = x_c(t)$ is defined following the dynamical system as

(18)
$$\frac{dx_c(t)}{dt} = u_c(t),$$
$$\frac{du_c(t)}{dt} = -\mathcal{S}_c \frac{p(x_c(t) + 0, t) - p(x_c(t) - 0, t)}{\mathcal{M}_c},$$

where $\mathcal{S}_c$ is the sectional area of the piston and $\mathcal{M}_c$ is the mass of the piston, the same as in (1). The pair of the initial position and velocity of the piston $(x_c(0), u_c(0)) = (0, u_0)$ is prescribed, and the pressures $p(x_c(t) \pm 0, t)$ will be given later. Compared to (1), the thickness L of the piston is assumed to be infinitesimal and is ignored.

Remark 3.1. This one-sided GRP can be formulated from the right-hand side $x_c(t) < x < \infty$ in the same way, thanks to the Galilean invariance.

Such a problem can be solved rigorously at least for a short time, following [36]. However, we are satisfied with the calculation of the instantaneous values

(13) $\mathbf{W}^*_{c,-} = \lim_{t \to 0+} \mathbf{W}(x_c(t) - 0, t), \quad \left(\frac{d\mathbf{W}}{dt}\right)^*_{c,-} = \lim_{t \to 0+} \frac{d_c}{dt} \mathbf{W}(x_c(t) - 0, t)$

for the implementation of the high order MBT method we propose. The same as the general GRP methodology [6], the calculation of (13) depends on solving the associated one-sided Riemann problem as formulated below.

Associated one-sided Riemann solver. In order to solve the above one-sided GRP (17), we first solve the one-sided Riemann problem following [8]. The associated one-sided Riemann problem is defined by approximating (17) with a constant initial data and neglecting the acceleration of the piston, which leads to the following IBVP:

(19)
$$\frac{\partial \mathbf{W}^A}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{W}^A)}{\partial x} = 0, \quad -\infty < x < x_c^A(t), t > 0,$$
$$\mathbf{W}^A(x, 0) = \mathbf{W}_-, \qquad -\infty < x < 0,$$
$$u^A(x_c^A(t), t) \equiv u_c(0), \qquad x_c^A(t) = u_c(0)t,$$

where $\mathbf{W}_-$ is regarded as a constant state. The boundary condition in (19) says that the piston moves with a uniform velocity, which alludes to the fact that no force is exerted on the piston. This is very different from the GRP solver that exhibits the acceleration effect.

The one-sided Riemann solver follows from the standard Riemann solver, e.g., in [14], to obtain the one-sided Riemann solution $(\mathbf{W}^A)^*_{c,-}$, by noting that the velocity of the one-sided Riemann solution is

(20) $(u^A)^*_{c,-} = u_c(0).$

The pressure $(p^A)^*_{c,-}$ exerted on the piston is obtained through the standard analysis for the Riemann solver, as figuratively shown in Figure 4.

One-sided GRP solver. The one-sided GRP (17) is asymptotically consistent with the one-sided Riemann problem (19) in the sense that

(21) $\mathbf{W}^*_{c,-} = (\mathbf{W}^A)^*_{c,-}.$

(a) The Riemann problem   (b) The classical piston problem
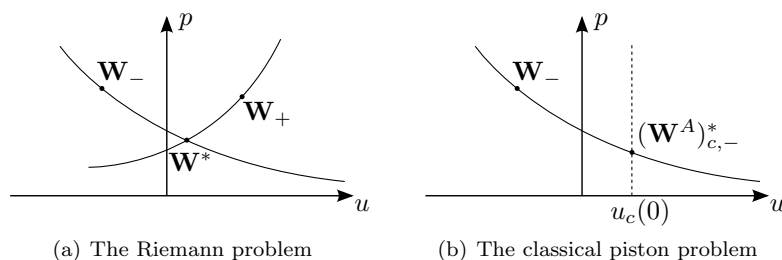
FIG. 4. *The state curves of the standard Riemann problem and the one-sided Riemann problem in the (u, p)-plane.*

Symmetrically, the one-sided GRP from the right-hand side, as formulated in (10), is associated with the right one-sided Riemann problem,

$$(22) \qquad \mathbf{W}^*_{c,+} = (\mathbf{W}^A)^*_{c,+},$$

where the annotation of obvious notation is suppressed. Note that the pressures from both sides are different, generally speaking

$$(23) \qquad p^*_{c,-} \neq p^*_{c,+}.$$

So the piston is accelerated, according to (18):

$$(24) \qquad \frac{du_c(t)}{dt}\bigg|_{t=0} = -\mathcal{S}_c \frac{p^*_{c,+} - p^*_{c,-}}{\mathcal{M}_c}.$$

This is the instantaneous acceleration of the piston for the one-sided GRP solver. Once we have this, we immediately use the standard GRP solver [6] to derive the relation

$$(25) \qquad a^*\Big(\frac{du}{dt}\Big)^*_{c,-} + b^*\Big(\frac{dp}{dt}\Big)^*_{c,-} = d^*,$$

where $\frac{d}{dt} = \frac{\partial}{\partial t} + u\frac{\partial}{\partial x}$, and the coefficients a^*, b^*, and d^* are determined from $\mathbf{W}_-(0)$, $\mathbf{W}'_-(0)$, and $\mathbf{W}^*_{c,-}$, as usually done for the GRP solver [6]. The procedure to calculate coefficients in (25) exactly follows that given in [6]. Readers are referred there for details.

According to the boundary condition in (17), we have $(\frac{du}{dt})^*_{c,-} = \frac{du_c}{dt}(0)$. Then $(\frac{dp}{dt})^*_{c,-}$ is obtained by solving the linear equation (25). As for the density, we have from the EOS, thanks to the fact $(ds/dt)^*_{c,-} = 0$, that

$$(26) \qquad \Big(\frac{d\rho}{dt}\Big)^*_{c,-} = \frac{1}{(c^*)^2}\Big(\frac{dp}{dt}\Big)^*_{c,-},$$

where c^* is the local sound speed defined by $c^* = \sqrt{\gamma p^*_{c,-}/\rho^*_{c,-}}$.

3.2. One-sided generalized Riemann problem solver in two dimensions. Think of a solid body in two dimensions with the initial translational and rotational velocities $\mathbf{u}_c(0)$ and $\omega(0)$. The pressure exerted on the surface determines the acceleration. Our strategy is made as follows: At any point on the surface, we will first solve

the one-sided Riemann problem normal to the surface to obtain the pressure, which yields the translational and rotational accelerations of the surface according to (5) and (6). Then we develop the two-dimensional one-sided GRP solver to obtain the instantaneous value of the derivative $d\mathbf{W}/dt$ for the flux evaluation and the boundary tracking.

Denote by $\partial\Omega(0)$ the initial position of the solid surface and recall the notation in (7). For any $\mathbf{X}_b \in \partial\Omega(0)$, denote by $\mathbf{x}_b(t; \mathbf{X}_b)$ the trajectory from $\mathbf{X}_b$ for $t > 0$. The unit normal vector of $\partial\Omega(t)$ at $\mathbf{x}_b(t; \mathbf{X}_b)$ is denoted by $\mathbf{n}(t; \mathbf{X}_b)$, and the instantaneous velocity of the solid surface at $\mathbf{x}_b(t; \mathbf{X}_b)$ is denoted by $\mathbf{u}_b(t; \mathbf{X}_b)$.

One-sided normal Riemann solver. For the sake of presentation simplicity, we place the section of the moving boundary we are concerned with initially along the y-axis, thanks to the Galilean invariance. Define the associated one-sided normal Riemann problem at any point $(0, y_*)$ on the surface of the solid body as

$$
(27) \quad
\begin{aligned}
&\frac{\partial \mathbf{W}^N}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{W}^N)}{\partial x} = 0, && -\infty < x < x_c^N(t), \ t > 0, \\
&\mathbf{W}^N(x, t = 0) = \mathbf{W}_-^N(0, y_*), && -\infty < x < 0, \\
&u^N(x_c^N(t), t) \equiv u_c^N, && x_c^N(t) = u_c^N \ t, \ t > 0,
\end{aligned}
$$

where the superscript "N" indicates the vector normal to the boundary $x = 0$. As far as the associated one-sided normal Riemann problem is concerned, the piston velocity in the IBVP (27) is defined as $u_c^N = \mathbf{u}_b(t = 0; 0, y_*) \cdot \mathbf{n}(t = 0; 0, y_*)$, where $\mathbf{u}_b(t = 0; 0, y_*)$ is given by the first equation in (7). The IBVP (27) can be solved in exactly the same way as the one-dimensional one-sided Riemann problem (19). The corresponding result of solving the above associated one-sided normal Riemann problem is denoted by $(\mathbf{W}^N)^*$.

One-sided GRP solver in two dimensions. Once we know the total pressure on the surface using the one-sided normal Riemann solver, we can calculate the acceleration according to (5) and (6), the numerical discretizations of which will be specified in subsection 4.2. Then we can solve the following two-dimensional one-sided GRP:

$$
(28) \quad
\begin{aligned}
&\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{W})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{W})}{\partial y} = 0, \\
&\mathbf{W}(x, y, 0) = \mathbf{W}_-(x, y), && x < 0, \ y \in \mathbb{R}, \\
&\mathbf{u}(\mathbf{x}_b(t; 0, y), t) \cdot \mathbf{n}(t; 0, y) = \mathbf{u}_b(t; 0, y) \cdot \mathbf{n}(t; 0, y), && y \in \mathbb{R}, \ t > 0,
\end{aligned}
$$

where, in light of (7), $\mathbf{x}_b(t; 0, y)$ is the trajectory of the boundary point starting from $(0, y)$ and its velocity is $\mathbf{u}_b(t; 0, y)$. The two-dimensional GRP solver serves to obtain the instantaneous values at any point $(0, y_*)$,

$$
(29) \quad \mathbf{W}_b^* = \lim_{t \to 0+} \mathbf{W}(\mathbf{x}_b(t; 0, y_*), t), \quad \left(\frac{d\mathbf{W}}{dt}\right)_b^* = \lim_{t \to 0+} \frac{d_c \mathbf{W}}{dt}(\mathbf{x}_b(t; 0, y_*), t),
$$

for any $y_* \in \mathbb{R}$ and particularly $y_* = 0$, where $\frac{d_c}{dt} = \frac{\partial}{\partial t} + u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y}$ is the directional derivative along the particle path. Recall again that $\mathbf{W}_b^*$ is just the one-sided normal Riemann solution of (27) and is given as $\mathbf{W}_b^* = (\mathbf{W}^N)^*$, which provides the local acceleration.

Different from the one-dimensional one-sided GRP solver, the transversal effect, as pointed out in [24], is important. Following [26], we regard the transversal term as a local source term and solve the following IBVP:

$$
\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{W})}{\partial x} = -\Big(\frac{\partial \mathbf{G}}{\partial y}\Big)^*,
$$

(30)
$$
\mathbf{W}(x, y, 0) = \mathbf{W}_-(x, y), \qquad\qquad x < 0, \ y \in \mathbb{R},
$$

$$
\mathbf{u}(\mathbf{x}_b(t; 0, y), t) \cdot \mathbf{n}(t; 0, y) = \mathbf{u}_b(t; 0, y) \cdot \mathbf{n}(t; 0, y), \quad y \in \mathbb{R}, \ t > 0,
$$

where $(\frac{\partial \mathbf{G}}{\partial y})^* = \frac{\partial \mathbf{G}}{\partial \mathbf{W}}(\mathbf{W}_b^*)(\frac{\partial \mathbf{W}}{\partial y})^*$ is a fixed value, as described in [26]. Then the two-dimensional one-sided GRP solver for (28) is derived exactly the same as the two-dimensional GRP solver in [5, 26].

4. High order moving boundary tracking algorithm. This section describes a high order MBT algorithm based on the newly developed one-sided GRP solver. This resulting scheme is different from those using the ghost fluid approach [22, 37], but can be regarded as the high order extension of MBT in [14].

4.1. One-dimensional high order MBT algorithm. The algorithm consists of three ingredients, as pointed out in section 2: *the evolution of flows, the moving boundary tracking, and the boundary cell merging.* The flow motion in the interior domain is treated using the standard finite volume method, e.g., the GRP method in [6]. So we focus on the boundary cell, which is denoted by $B_J(t_n, t_{n+1}) = \{(x,t); x \in (x_{J-\frac{1}{2}}, x_c(t)), t_n \le t \le t_{n+1}\}$. The algorithm is described as follows:

(i) *Implementation of one-sided GRP solver.* Given the initial data over the boundary cell I_J^c at $t = t_n$,

$$
(31) \qquad \mathbf{W}_J^{c,n}(x) = \overline{\mathbf{W}}_J^{c,n} + \Big(\frac{\partial \mathbf{W}}{\partial x}\Big)_J^{c,n}\Big(x - \frac{x_{J-\frac{1}{2}} + x_c(t_n)}{2}\Big),
$$

we use the one-sided GRP solver to obtain the instantaneous values $(\mathbf{W}_{c,-}^{n,*}, (\frac{d\mathbf{W}}{dt})_{c,-}^{n,*})$, which is annotated in (13). Then we compute the midpoint values for the flux approximation and interface values for slope evaluation at the next time level,

$$
(32) \quad \mathbf{W}_{c,-}^{n+\frac{1}{2}} = \mathbf{W}_{c,-}^{n,*} + \frac{\Delta t}{2}\Big(\frac{d\mathbf{W}}{dt}\Big)_{c,-}^{n,*}, \quad \mathbf{W}_{c,-}^{n+1} = \mathbf{W}_{c,-}^{n,*} + \Delta t\Big(\frac{d\mathbf{W}}{dt}\Big)_{c,-}^{n,*}.
$$

(ii) *The tracking of the moving boundary.* We track the moving boundary to the next time level using the formula within the second order accuracy,

$$
(33) \qquad\qquad x_c(t_{n+1}) = x_c(t_n) + \Delta t u_c^n + \frac{\Delta t^2}{2}\Big(\frac{du_c}{dt}\Big)^n.
$$

(iii) *Flow evolution.* We use the finite volume framework (12) to evolve the flow,
(34)
$$
|I_J^c(t_{n+1})|\overline{\mathbf{W}}_J^{c,n+1} = |I_J^c(t_n)|\overline{\mathbf{W}}_J^{c,n} - \Delta t\left[\Big(\mathbf{F}_{c,-}^{n+\frac{1}{2}} - u_c^{n+\frac{1}{2}}\mathbf{W}_{c,-}^{n+\frac{1}{2}}\Big) - \mathbf{F}_{J-\frac{1}{2}}^{n+\frac{1}{2}}\right],
$$

where $I_J^c(t_{n+1}) = (x_c(t_{n+1}) - x_{J-\frac{1}{2}})$, $\overline{\mathbf{W}}_J^{c,n+1}$ is the average of $\mathbf{W}(x, t_{n+1})$ over $I_J^c(t_{n+1})$, and

$$
(35) \qquad\qquad \mathbf{F}_{c,-}^{n+\frac{1}{2}} = \mathbf{F}(\mathbf{W}_{c,-}^{n+\frac{1}{2}}), \qquad \mathbf{F}_{J-\frac{1}{2}}^{n+\frac{1}{2}} = \mathbf{F}(\mathbf{W}_{J-\frac{1}{2}}^{n+\frac{1}{2}}).
$$

The half-time value $\mathbf{W}_{J-\frac{1}{2}}^{n+\frac{1}{2}}$ is defined by $\mathbf{W}_{J-\frac{1}{2}}^{n+\frac{1}{2}} = \mathbf{W}_{J-\frac{1}{2}}^{n,*} + \frac{\Delta t}{2}(\frac{\partial \mathbf{W}}{\partial t})_{J-\frac{1}{2}}^{n,*}$, where $\mathbf{W}_{J-\frac{1}{2}}^{n,*}$ and $(\frac{\partial \mathbf{W}}{\partial t})_{J-\frac{1}{2}}^{n,*}$ are obtained by applying the standard GRP solver at $x_{J-\frac{1}{2}}$ [6].

(iv) *Cell-merging procedure.* Obey the one-dimensional cell-merging criterion to construct a "legal" boundary cell at the new time level $t = t_{n+1}$.

(v) *Data update.* Update the slope over the boundary cell $I_J^c(t_{n+1})$,

$$(36) \quad \left(\frac{\partial \mathbf{W}}{\partial x}\right)_J^{c,n+1} = \text{minmod}\left(\frac{\mathbf{W}_{c,-}^{n+1} - \overline{\mathbf{W}}_{J-\frac{1}{2}}^{n+1}}{|I_J^c(t_{n+1})|}, \alpha\frac{\overline{\mathbf{W}}_J^{c,n+1} - \overline{\mathbf{W}}_{J-1}^{n+1}}{(\Delta x + |I_J^c(t_{n+1})|)/2}\right),$$

where $\alpha \in (0, 2)$ is a user-defined parameter, and taken as 1.9 in the present paper in order to make the profile of the data $\mathbf{W}^{n+1}(x)$ sawtooth and non-oscillatory essentially. The minmod function is taken componentwise. This is the same reconstruction approach as in [4, 6]. Then we reconstruct the data in the form (31) over the boundary cell $I_J^c(t_{n+1})$.

4.2. Two-dimensional high order MBT algorithm. The two-dimensional MBT algorithm has the same methodology as the one-dimensional case; in addition, the moving boundary tracking is more technical. Therefore we only provide details for boundary tracking and the data reconstruction by assuming that all necessary values are available from the two-dimensional one-sided GRP solver. We still use the Cartesian mesh as expressed in (14). At t_n, denote by I_μ^c the cells cut by $\Omega(t_n)$ for $\mu = 1, 2, \ldots, M$. Denote the segment of $\partial\Omega(t_n)$ inside I_μ^c by Γ_μ^n with the middle point $\mathbf{x}_\mu^n$.

(i) *Two-dimensional moving boundary tracking.* In light of (7), the instantaneous velocity at the middle point $\mathbf{x}_\mu^n$ of Γ_μ^n is given by

$$(37) \quad \mathbf{u}_b(\mathbf{x}_\mu^n, t_n) = \mathbf{u}_c(t_n) + \omega(t_n)\left[\mathbf{x}_\mu^n - \mathbf{x}_c(t_n)\right]^{\perp}.$$

The associated one-sided normal Riemann problem (27) is consequently defined at $\mathbf{x}_\mu^n$. Denote the corresponding one-sided Riemann solution by $\mathbf{W}_{\mathbf{x}_\mu^n}^{n,*}$. Particularly, the boundary pressure $p_{\mathbf{x}_\mu^n}^{n,*}$ is obtained. Therefore the translational and rotational accelerations of $\Omega(t_n)$ are, according to (5) and (6),

$$(38) \quad \begin{aligned}\frac{d\mathbf{u}_c}{dt}(t_n) &= -\frac{1}{\mathcal{M}_c}\sum_\mu p_{\mathbf{x}_\mu^n}^{n,*}\,|\Gamma_\mu^n|\,\mathbf{n}_\mu^n, \\ \frac{d\omega}{dt}(t_n) &= \frac{1}{\mathcal{A}_c}\sum_\mu\left\{p_{\mathbf{x}_\mu^n}^{n,*}\,|\Gamma_\mu^n|\,\left[\mathbf{n}_\mu^n \times (\mathbf{x}_\mu^n - \mathbf{x}_c(t_n))\right]\right\}.\end{aligned}$$

Consequently, in light of (7), the acceleration of the point $\mathbf{x}_\mu^n \in \partial\Omega(t_n)$ is

$$(39) \quad \begin{aligned}\frac{d\mathbf{u}_b}{dt}(\mathbf{x}_\mu^n, t_n) &= \frac{d\mathbf{u}_c}{dt}(t_n) + \frac{d\omega}{dt}(t_n)\left[\mathbf{x}_\mu^n - \mathbf{x}_c(t_n)\right]^{\perp} \\ &\quad + \omega(t_n)\left\{\omega(t_n)\left[\mathbf{x}_\mu^n - \mathbf{x}_c(t_n)\right]^{\perp}\right\}^{\perp}.\end{aligned}$$

Since the midpoint value is used on Γ_μ^n and $|\Gamma_\mu^n| \sim \Delta x$, the quadrature (38) is second order accurate in space so that the approximation of translational and

rotational accelerations in (38) as well as the instantaneous acceleration in (39) are of order $\mathcal{O}(\Delta x^2)$. Using the one-sided GRP solver, we can obtain the derivatives of the solution $\mathbf{W}(\mathbf{x}, t)$ along the moving boundary. It turns out that (38) is sufficient for the algorithm to achieve second order accuracy.

The above procedure of tracking the moving boundary provides the information of the piston motion required by the two-dimensional one-sided GRP solver to solve (30) in section 3.2. At last, we update relevant values to the next time level as

$$\mathbf{x}_c(t_{n+1}) = \mathbf{x}_c(t_n) + \Delta t \mathbf{u}_c(t_n) + \frac{\Delta t^2}{2}\frac{d\mathbf{u}_c}{dt}(t_n),$$

$$(40) \qquad \theta(t_{n+1}) = \theta(t_n) + \Delta t \omega(t_n) + \frac{\Delta t^2}{2}\frac{d\omega}{dt}(t_n),$$

$$\mathbf{u}_c(t_{n+1}) = \mathbf{u}_c(t_n) + \Delta t \frac{d\mathbf{u}_c}{dt}(t_n), \quad \omega(t_{n+1}) = \omega(t_n) + \Delta t \frac{d\omega}{dt}(t_n).$$

(ii) *Data reconstruction over boundary cells.* For the cut cell $I_\mu^c(t_{n+1})$, denote all of its edges by $\Gamma_\mu^{(k)}(t_{n+1})$ for $k = 0, 1, \ldots, K$. Furthermore, assume that $\Gamma_\mu^{(0)}(t_{n+1}) = \partial\Omega(t_{n+1}) \cap I_\mu(t_{n+1})$. The unit outer normal vectors of $\Gamma_\mu^{(k)}(t_{n+1})$ with respect to $I_\mu^c(t_{n+1})$ are denoted by $\mathbf{n}_\mu^{(k)}(t_{n+1})$. In order to linearly approximate $\mathbf{W}(\mathbf{x}, t_{n+1})$ in $I_\mu^c(t_{n+1})$, we need to estimate its gradient. By the Gauss theorem, we have

$$(41) \quad \int_{I_\mu^c(t_{n+1})} \nabla\mathbf{W}(\mathbf{x}, t_{n+1})d\mathbf{x} = \sum_{k=0}^{K} \int_{\Gamma_\mu^{(k)}(t_{n+1})} \mathbf{W}(\mathbf{x}, t_{n+1}) \otimes \mathbf{n}(\mathbf{x}, t_{n+1})dl.$$

To estimate the gradient of $\mathbf{W}$ in $I_\mu^c(t_{n+1})$, we discretize (41) as

$$(42) \qquad (\nabla\mathbf{W})_\mu^{n+1} = \frac{1}{|I_\mu^c(t_{n+1})|} \sum_{k=0}^{K} |\Gamma_\mu^{(k)}(t_{n+1})| \mathbf{W}_\mu^{(k),n+1} \otimes \mathbf{n}_\mu^{(k),n+1},$$

where $\mathbf{W}_\mu^{(0),n+1} = \mathbf{W}_\mu^{n,*} + \Delta t (\frac{d\mathbf{W}}{dt})_\mu^{n,*}$, and $\mathbf{W}_\mu^{n,*}$ and $(\frac{d\mathbf{W}}{dt})_\mu^{n,*}$ are obtained by solving the two-dimensional one-sided GRP at $(\mathbf{x}_\mu^n, t^n)$. The least squares limiter, which is a multidimensional version of (36), can be used to suppress possible oscillations [3].

5. Accuracy analysis. In this section we analyze the accuracy of the proposed MBT method. We just focus on "legal" boundary cells $I_j^c(t)$. Recall the flux approximation in (35). Then we estimate

$$(43) \qquad \mathbf{F}_{J-\frac{1}{2}}^{n+\frac{1}{2}} - \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \mathbf{F}(\mathbf{W}(x_{J-\frac{1}{2}}, t))dt = -\frac{\Delta t^2}{6}\frac{\partial^2\mathbf{F}}{\partial t^2}(x_{J-\frac{1}{2}}, t_n) + \mathcal{O}(\Delta t^3)$$

and
(44)

$$\left[\mathbf{F}_{c,-}^{n+\frac{1}{2}} - u_c^{n+\frac{1}{2}}\mathbf{W}_{c,-}^{n+\frac{1}{2}}\right] - \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}}[\mathbf{F}(\mathbf{W}(x_c(t) - 0, t)) - u_c\mathbf{W}(x_c(t) - 0, t)]dt$$

$$= -\frac{\Delta t^2}{6}\frac{d_c^2}{dt^2}[\mathbf{F}(\mathbf{W}) - u\mathbf{W}](x_c(t_n) - 0, t_n) + \mathcal{O}(\Delta t^3).$$

The flux difference yields

(45)
$$\frac{1}{\Delta t} \left\{ \int_{t_n}^{t_{n+1}} \left[\mathbf{F}(\mathbf{W}(x_c(t)-0,t)) - u_c(t)\mathbf{W}(x_c(t)-0,t) \right] dt - \int_{t_n}^{t_{n+1}} \mathbf{F}(\mathbf{W}(x_{J-\frac{1}{2}},t)) dt \right\}$$
$$- \left[\left(\mathbf{F}_{c,-}^{n+\frac{1}{2}} - u_c^{n+\frac{1}{2}} \mathbf{W}_{c,-}^{n+\frac{1}{2}} \right) - \mathbf{F}_{J-\frac{1}{2}}^{n+\frac{1}{2}} \right]$$
$$= \frac{\Delta t^2}{6} \left[\frac{d_c^2}{dt^2} (\mathbf{F}(\mathbf{W}) - u\mathbf{W})(x_c(t_n) - 0, t_n) - \frac{\partial^2}{\partial t^2} \mathbf{F}(\mathbf{W})(x_{J-\frac{1}{2}}, t_n) \right] + \mathcal{O}(\Delta t^3).$$

It is evident that the time varying of boundary cells introduces extra errors. If the boundary is fixed, i.e., $u_c \equiv 0$, we can achieve the second order accuracy for $\Delta t \sim \Delta x$, which is written as

(46)
$$\overline{\mathbf{W}}_c^{n+1} - \frac{1}{|I_J^c(t_{n+1})|} \int_{I_J^c(t_{n+1})} \mathbf{W}(x, t_{n+1}) dx = \mathcal{O}(\Delta t^3).$$

Otherwise the error has the form

(47)
$$\overline{\mathbf{W}}_c^{n+1} - \frac{1}{|I_J^c(t_{n+1})|} \int_{I_J^c(t_{n+1})} \mathbf{W}(x, t_{n+1}) dx = \mathcal{O}(\Delta t^3/\Delta x).$$

Up to a fixed time, the number of computational steps is $\mathcal{O}(\Delta t^{-1})$. Thus the accumulated error in the boundary control volume is $\mathcal{O}(\Delta t^2/\Delta x)$. This implies that second order accuracy is sought only if $\Delta t \sim \Delta x^{3/2}$.

We use the example proposed in [16] to verify the accuracy of the current MBT method. This example describes a tube filled with gas initially,

$$\begin{bmatrix} \rho_0(x) \\ u_0(x) \\ p_0(x) \end{bmatrix} = \begin{bmatrix} 1 + 0.2\cos(2\pi x) \\ 2(x+0.5)\, u_c(0) \\ [\rho_0(x)]^\gamma \end{bmatrix}, \quad x \in (-0.5, 0),$$

where $\gamma = 1.4$ and $u_c(0)$ will be specified later. The left and right ends of the tube are closed by a fixed wall and a moving piston, respectively. Two cases are considered. The first case has a uniform piston velocity $u_c(t) \equiv -0.5$, while the second case has a varying piston velocity $u_c(t) = -2t$. Computations stop at $t = 0.5$ and the piston is located at $x = -0.75$ for both cases.

Numerical results shown below are obtained by the second order MBT scheme, using the regular CFL condition

(48)
$$\Delta t = \mathrm{CFL} \frac{\Delta x}{\lambda_{\max}}$$

and the modified CFL condition

(49)
$$\Delta t = \mathrm{CFL} \frac{\Delta x^{\frac{3}{2}}}{\lambda_{\max}},$$

where $\lambda_{\max}$ is the maximum signal speed. The CFL number is taken to be 0.6. Following the recommendation from [14], we set the user-tuned parameter to be $\kappa = 0.3$. Since the entropy is uniform for both cases, it is chosen for comparison. As shown in Table 1 for the first case and in Table 2 for the second, the current MBT with the acceleration element achieves the accuracy of desired order under the modified CFL condition (49).
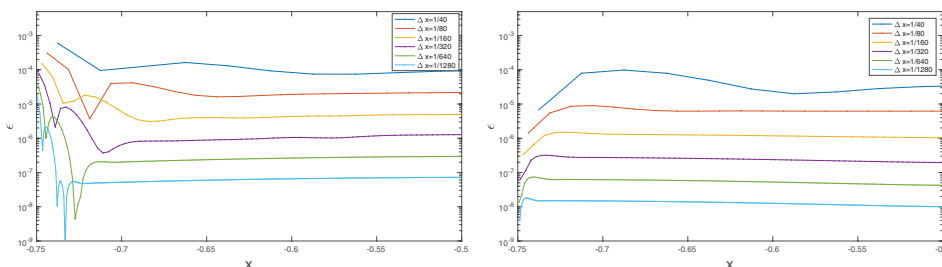
TABLE 1

Entropy errors and convergence rates of the one-dimensional accuracy test. The piston moves with a uniform velocity $u_c(t) \equiv -0.5$. The second order MBT scheme is performed using the regular CFL condition (48) and the modified CFL condition (49), respectively. $CFL = 0.6$ and $\kappa = 0.3$.

Mesh	Regular CFL condition (48)			
size	L_1 errors	L_1 orders	L_∞ errors	L_∞ orders
1/40	4.35e-05		5.95e-04	
1/80	9.58e-06	2.18	3.06e-04	0.96
1/160	2.29e-06	2.06	1.52e-04	1.01
1/320	5.63e-07	2.03	7.46e-05	1.02
1/640	1.47e-07	1.94	3.61e-05	1.05
1/1280	3.86e-08	1.93	1.72e-05	1.07
Mesh	Modified CFL condition (49)			
size	L_1 errors	L_1 orders	L_∞ errors	L_∞ orders
1/40	1.91e-05		9.81e-05	
1/80	2.58e-06	2.89	9.00e-06	3.45
1/160	4.01e-07	2.69	1.50e-06	2.58
1/320	7.32e-08	2.45	3.19e-07	2.24
1/640	1.56e-08	2.23	7.36e-08	2.12
1/1280	3.68e-09	2.08	1.80e-08	2.03

FIG. 5. *Entropy errors of the one-dimensional accuracy test where the piston moves with a varying velocity $u_c(t) \equiv -0.5$. The second order MBT scheme is performed using the regular CFL condition (48) and the modified condition (49), respectively.*

In order to see clearly the influence of errors resulting from the deterioration of accuracy over cells neighboring to the moving boundary under the regular CFL condition (48), we display the entropy error $\epsilon(x) = |s(x, 0.5) - 1|$ in Figures 5 and 6 for both cases, respectively. The errors are shown in the region $x \in (-0.75, -0.5)$ neighboring to the moving boundary. The local truncation errors are large under condition (48) and can be eliminated using the modified CFL condition (49) so that second order accuracy is achieved.

Remark 5.1. We have carried out extensive numerical experiments for $\kappa \in (0.3, 0.5)$ and observe that the user-tuned parameter κ has no influence on the numerical performance. Therefore there is no need to list the results here.

For comparison, we mention the first order MBT scheme proposed in [14]. We perform an analysis in analogy with (43)–(47) to indicate that the local errors over the boundary control volume is accumulated at some fixed time to be $\mathcal{O}(\Delta t / \Delta x)$ under the regular CFL condition (48), which implies the fact that the error does not tend to zero as the mesh size is refined. We display the results in Table 3. One possible remedy for this is to make a constraint on the time step length $\Delta t \sim \Delta x^2$, similar to (49).

Next we verify the accuracy of the two-dimensional MBT scheme using the test

TABLE 2
Entropy errors and convergence rates of the one-dimensional accuracy test. The piston moves with a varying velocity $u_c(t) = -2t$. The second order MBT scheme is performed using the regular CFL condition (48) and the modified CFL condition (49), respectively. CFL $= 0.6$ and $\kappa = 0.3$.

| Mesh	Regular CFL condition (48)			
size	L_1 errors	L_1 orders	L_∞ errors	L_∞ orders
1/40	1.10e-04		1.01e-03	
1/80	2.28e-05	2.27	6.18e-04	0.71
1/160	5.62e-06	2.02	3.22e-04	0.94
1/320	1.33e-06	2.08	1.58e-04	1.02
1/640	3.18e-07	2.07	7.62e-05	1.05
1/1280	7.85e-08	2.02	3.58e-05	1.09
Mesh	Modified CFL condition (49)			
size	L_1 errors	L_1 orders	L_∞ errors	L_∞ orders
1/40	1.27e-04		1.42e-03	
1/80	2.07e-05	2.61	2.45e-04	2.53
1/160	3.99e-06	2.38	3.79e-05	2.69
1/320	8.20e-07	2.28	8.51e-06	2.16
1/640	1.65e-07	2.31	1.80e-06	2.24
1/1280	3.37e-08	2.30	3.27e-07	2.46
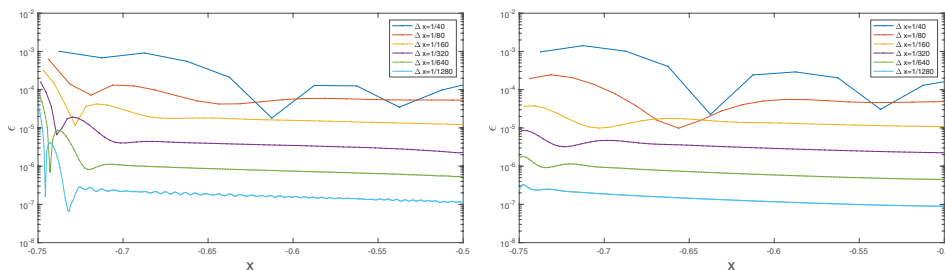
FIG. 6. *Entropy errors of the one-dimensional accuracy test where the piston moves with a varying velocity $u_c(t) = -2t$. The second order MBT scheme is performed using the regular and the modified CFL conditions, respectively.*

TABLE 3
Entropy errors and convergence rates of the one-dimensional accuracy test, using the first order MBT scheme and the regular CFL condition (48). The piston moves with a uniform velocity $u_c(t) \equiv -0.5$ and a varying velocity $u_c(t) = -2t$, respectively. CFL $= 0.6$ and $\kappa = 0.3$.

Mesh	Uniform piston velocity			
size	L_1 errors	L_1 orders	L_∞ errors	L_∞ orders
1/20	1.40e-03		3.24e-03	
1/40	7.11e-04	0.98	1.61e-03	1.01
1/80	3.39e-04	1.07	1.08e-03	0.57
1/160	1.63e-04	1.05	1.68e-03	-0.63
1/320	9.18e-05	0.83	5.68e-03	-1.76
1/640	1.80e-04	-0.97	3.50e-02	-2.62
Mesh	Varying piston velocity			
size	L_1 errors	L_1 orders	L_∞ errors	L_∞ orders
1/20	4.13e-03		6.03e-02	
1/40	2.50e-03	0.72	6.53e-02	-0.12
1/80	1.36e-03	0.87	6.54e-02	-0.00
1/160	7.35e-04	0.89	7.08e-02	-0.11
1/320	4.50e-04	0.71	1.15e-01	-0.70
1/640	4.45e-04	0.02	1.42e-01	-0.30

case proposed in [37]. The initial computational domain is $\{(x, y) : 0.5 < x < 1, 0 < y < 1\}$. The top, bottom, and right boundaries are all fixed reflective walls. The left boundary of the computational domain is a moving wall, whose position is determined by $x_c(t) = 0.5 - 0.5 \sin(t)$. The initial data is given as

$$
\begin{bmatrix}
\rho_0(x,y) \\
u_0(x,y) \\
v_0(x,y) \\
p_0(x,y)
\end{bmatrix}
=
\begin{bmatrix}
1 + 0.2 \cos(2\pi x) + 0.1 \cos(2\pi y) \\
x - 1 \\
y(1-y)\cos(\pi x) \\
[\rho_0(x,y)]^{\gamma}
\end{bmatrix}
, \quad 0.5 < x < 1, \ 0 < y < 1,
$$

where $\gamma = 1.4$. The flow is simulated to the time $T = 0.2$ with 20×20 up to 320×320 computational cells. As with the one-dimensional cases above, the entropy is uniform. The L_1 and L_∞ errors about the entropy and the corresponding numerical orders are tabulated in Table 4, which shows that the MBT scheme attains the expected second order accuracy.

TABLE 4

Entropy errors and convergence rates of the two-dimensional accuracy test. $CFL = 0.6$ and $\kappa = 0.3$.

Mesh size	L_1 errors	L_1 orders	L_∞ errors	L_∞ orders
1/20	7.63e-5		4.11e-4	
1/40	1.63e-5	2.22	1.01e-4	2.02
1/80	3.68e-6	2.15	2.42e-5	2.06
1/160	8.69e-7	2.08	5.71e-6	2.08
1/320	2.13e-7	2.03	1.58e-6	1.85

6. Numerical experiments. In this section, we will present several typical examples to demonstrate the performance of the current MBT method. The first two examples are one-dimensional, while the last two are two-dimensional. These examples are already benchmark problems in this field, and we refer to them as such. For all four cases, the CFL number is taken to be 0.6, the user-tuned parameter κ is taken as 0.3, and the parameter α for the minmod function in (36) is taken as 1.9.

6.1. Sudden motion of a piston with a rarefaction wave and a shock. This benchmark test is taken from [35]. An infinitely long tube is filled with gas, initially having the state $\rho_0 = 1$, $p_0 = 5/7$, $u_0 = 0$, and $\gamma = 1.4$. The computational domain is $[-20, 50]$, divided into equally distributed cells with the spatial size $\Delta x = 0.25$. A piston of width 2.5 is initially centered at $x_c = -1.25$. The piston suddenly moves to the right with a constant speed $u_c = 2$. Then a shock and a rarefaction wave forms ahead and behind the piston, respectively. The computation is performed up to the time $T = 11$. The numerical results of the density and pressure are shown in Figure 7, and both of them match well with the exact solution.

6.2. Sod shock interacting with a piston. We take this example from [20]. A tube occupies the space interval $[0, 3]$, filled with gas at rest initially,

$$
(\rho, u, p)(x, 0) = \begin{cases} (1, \ 0, \ 1), & x < 1, \\ (0.125, \ 0, \ 0.1), & x > 1. \end{cases}
$$

The specific heat ratio of the gas is $\gamma = 1.4$. Both ends of the tube are rigid walls. The computational domain $[0, 3]$ is uniformly divided into 600 cells. This tube is separated by a piston with thickness 0.2, centered initially at 1.5. The mass of the
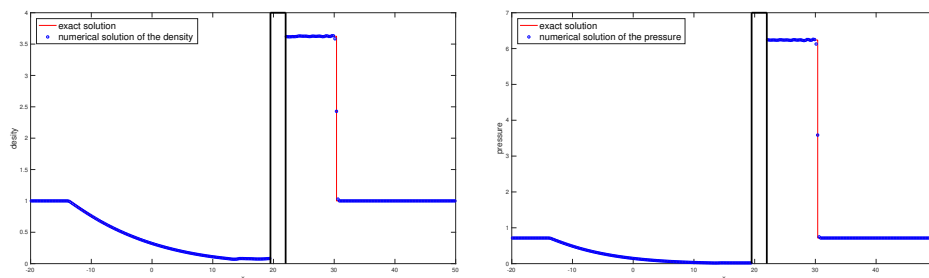
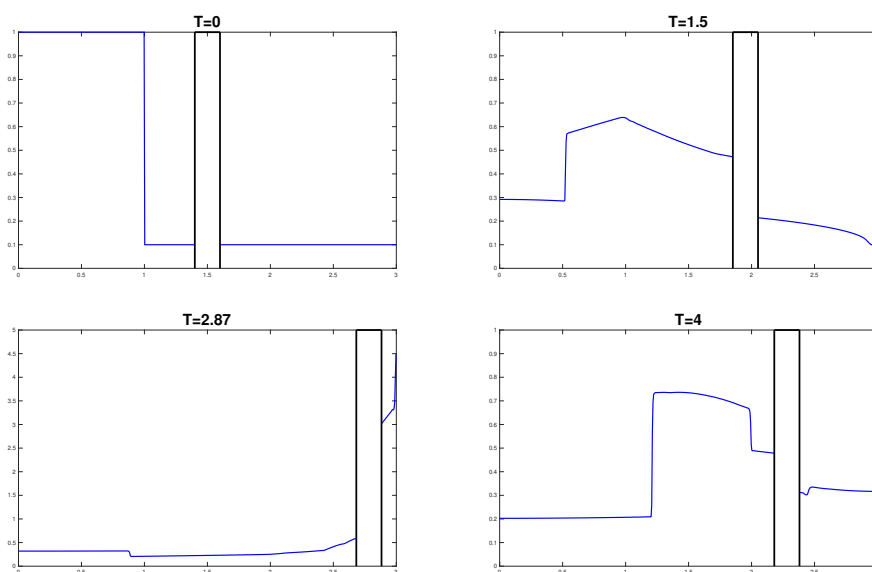FIG. 7. *The density (left) and the pressure (right) profiles at $T = 11$ for the sudden motion of a piston.*

FIG. 8. *The piston position and the pressure profile for the Sod shock interacting with a piston at $T = 0$, $T = 1.5$, $T = 2.87$, and $T = 4$.*

piston is $\mathcal{M}_\Omega = 1.0$. The motion of the piston passively depends on the force exerted from the fluid.

The fluid flow is described as follows: A shock from the left pushes the piston to the right. Since the right end of the tube is closed, the gas is compressed and the pressure increases as the piston moves right. Finally the piston is bounced back from the right. Figure 8 shows snapshots of the pressure of the flow field and the piston positions at different times $T = 0, 1.5, 2.87, 4$, respectively. The results match well with those in [20, Figure 12]. In addition, the MBT scheme is carried out on a series of refined grids, and the positions of the piston at $T = 1.5$ are compared with a reference value $x_c = 1.951855$ by a computation using 15000 cells. The errors and the convergence rates are listed in Table 5.

6.3. The kicking off of two-dimensional objects. We consider a cylinder, an oval disk, and a rectangular object interacting with a shock, respectively.

TABLE 5
The errors of the piston position at $T = 1.5$ and the convergence rates.

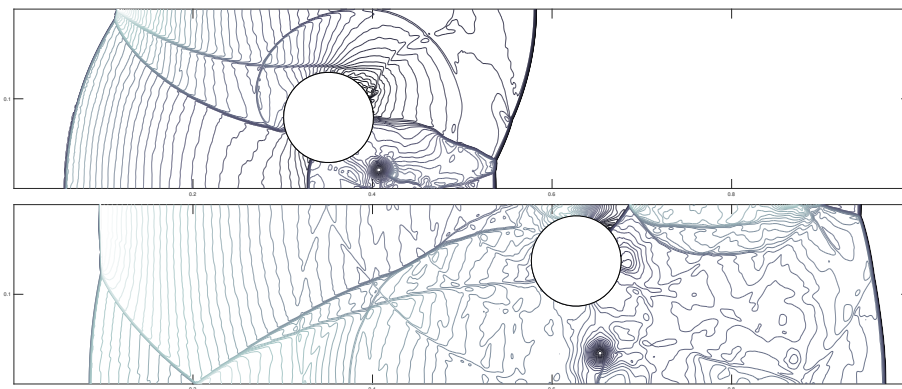| Δx	1/50	1/100	1/200	1/400	1/800
Position error	2.40e-3	1.44e-3	6.96e-4	3.21e-4	1.60e-4
Convergence rates		0.73	1.06	1.11	1.01

FIG. 9. *The position of the cylinder and the pressure of the flow field at $T = 0.1641$ (upper) and $T = 0.30085$ (lower).*

TABLE 6
Positions of the barycenter of the cylinder at $T = 0.30085$ as the mesh is refined.

Mesh size	1/400	1/500	1/600	1/800	1/1000
x coordinate	0.6213	0.6240	0.6247	0.6261	0.6269
y coordinate	0.1330	0.1345	0.1355	0.1367	0.1370

The first case considered here is one that has been widely used [14, 16, 1, 22, 20, 27, 31, 29, 9]. A tunnel occupies the region $[0, 1] \times [0, 0.2]$, whose top and bottom boundaries are both reflective walls. A shock of Mach number 3 is posed at $x = 0.08$ and moves right. The preshock state is given as $(\rho_0, u_0, v_0, p_0) = (1.4, 0, 0, 1)$. The specific heat ratio of the gas is $\gamma = 1.4$. A cylinder with the radius $r = 0.05$ is placed on the ground with its center located at $(0.15, 0.05)$. The density of the cylinder is $\rho_\Omega = 10.77$, which makes its mass $\mathcal{M}_\Omega = \pi r^2 \rho_\Omega$ and inertia $\mathcal{A}_\Omega = \frac{\mathcal{M}_\Omega}{2} r^2$. As the shock moves to the right, the cylinder would be kicked off from the ground. Figure 9 shows the cylinder positions and the pressure profiles at $T = 0.1641$ (upper panel) and $T = 0.30085$ (lower panel) using 1000×200 computational cells. Slightly different from the conclusion in [22], a vortex beneath the cylinder is observed as it moves, which supports observations in [16, 27, 31, 29, 9].

The positions of the cylinder barycenter at $T = 0.30085$ with an increasing resolution are listed in Table 6. The horizontal positions match well the results in [1] and [9] by the GoHy-3 scheme. The vertical positions by the current MBT scheme are lower than those in [1, 9] and other aforementioned papers. Comparison of the barycenter position with the numerical results in [1, 9] is shown in Figure 10. Nevertheless, it is observed that the final position of the cylinder by the current MBT scheme has a good qualitative agreement with that in [20, Figure 16].

Next, consider the kicking off of an oval disk. In the tunnel $[0, 1] \times [0, 0.2]$, a Mach 3 shock is initially placed at $x = 0.08$. Following the original setup in [4], the
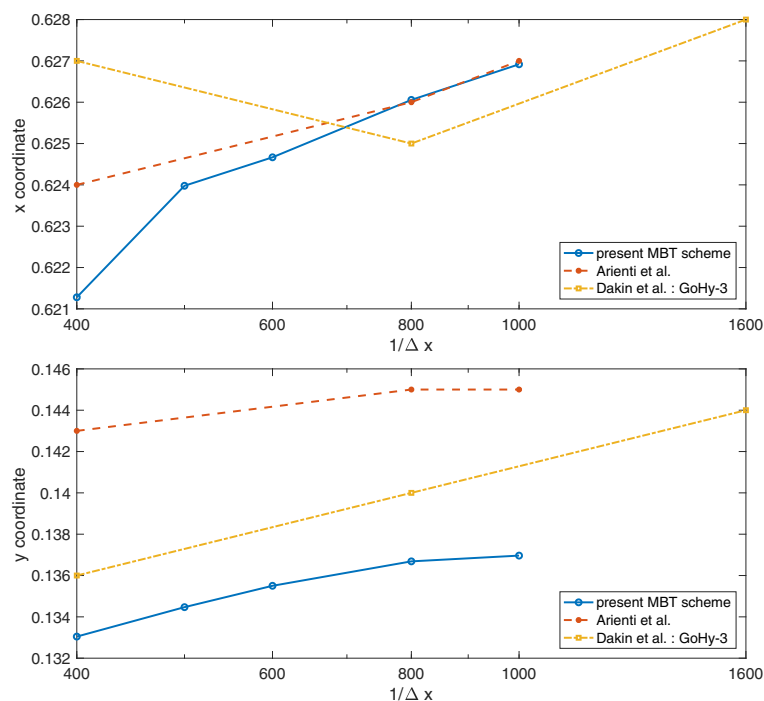
FIG. 10. *Comparison of the horizontal (upper) and vertical (lower) positions of the cylinder barycenter with those in Arienti et al.* [1] *and Dakin, Després, and Jaouen* [9].

preshock state of the gas is $(\rho_0, u_0, v_0, p_0) = (1.3, 0, 0, 0.1)$. The barycenter of the oval disk is initially placed at $(0.15, 0.03)$. The two axes of the oval disk are $a = 0.0625$ and $b = 0.025$, respectively. The long axis of the oval disk is placed horizontally. The density of the disk is $\rho_\Omega = 10\rho_0$. The mass and the inertia of the disk are given by $\mathcal{M}_\Omega = \pi a b \rho_\Omega$ and $\mathcal{A}_\Omega = \frac{\mathcal{M}_\Omega}{4}(a^2 + b^2)$. As the shock moves right, the disk would be kicked off from the ground and rotates as it moves in the flow field. The numerical computation is performed over 800×160 cells. Figure 11 shows the positions of the disk and the pressure profiles of the flow field at $T = 0.6$ (upper panel) and $T = 1.0$ (lower panel). Multiple reflected shocks and vortices formed beneath and near the disk can be observed. The numerical results match those proposed in [4] perfectly and with better resolutions.

Finally, we replace the oval disk by a rectangular object whose barycenter initially locates at $(0.15, 0.06)$. The long and short sides of the rectangle are $a = 0.06$ and $b = 0.03$, respectively. The rectangle is placed in a manner such that its long side makes an angle of $\pi/4$ with the negative x-direction. The mass and the inertia of the rectangle are $\mathcal{M}_\Omega = ab\rho_\Omega$ and $\mathcal{A}_\Omega = \frac{\mathcal{M}_\Omega}{3}(a^2 + b^2)$, where $\rho_\Omega = 10\rho_0$. As the shock moves right, the rectangular object is kicked off. A computation is performed over a 800×160 grid. The positions of the rectangle and the pressure profiles of the flow field are displayed in Figure 12, at $T = 0.6$ (upper panel) and $T = 1.0$ (lower panel). Compared with the oval disk, a rectangular object has a larger inertia with respect to its size. Therefore it rotates less, as shown numerically.

Remark 6.1. For this case, the presence of right angles on the moving boundary leads to L-shaped cut cells, which are ill-shaped. This requires a more delicate cell-
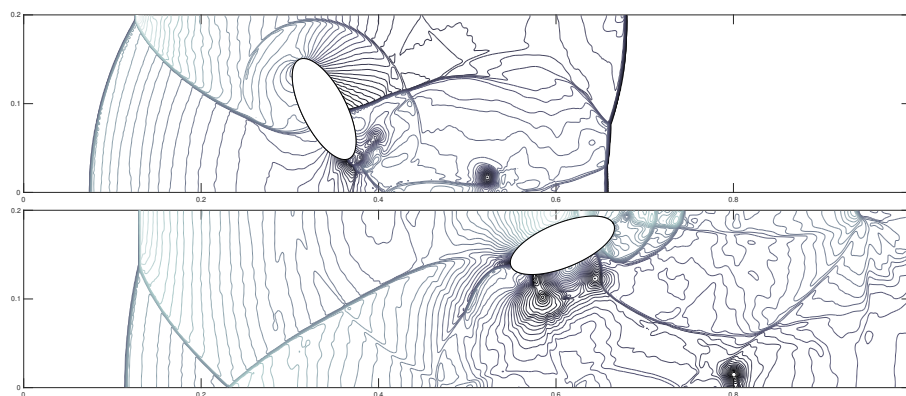
FIG. 11. *The positions of the oval disk and the pressure profiles of the flow field at $T = 0.6$ (upper) and $T = 1.0$ (lower).*
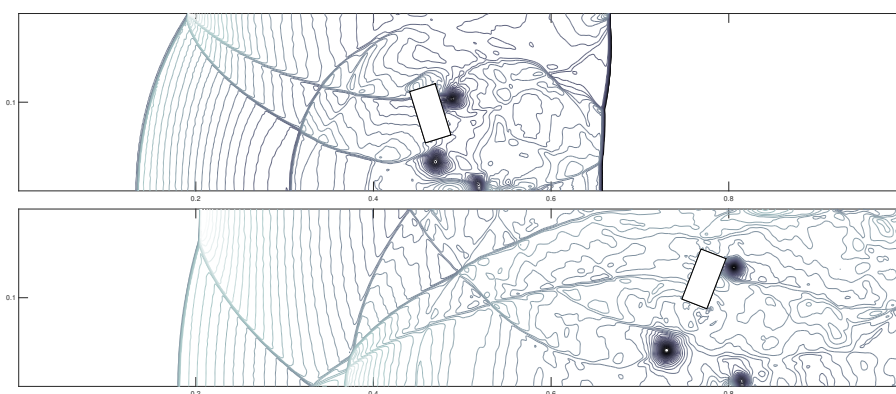

FIG. 12. *The positions of the rectangular object and the pressure profiles of the flow field at $T = 0.6$ (upper) and $T = 1.0$ (lower).*
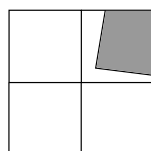

FIG. 13. *The cell-merging procedure for an L-shaped cut cell.*

merging procedure. In this paper we merge the cut cell with more neighbors to form a larger control volume, as shown in Figure 13. The AMR approach in [33] could be a promising solution if larger errors are produced due to the relatively large control volume.

To conclude this subsection, we simulate the kicking off of the oval disk and the rectangle over a series of refined grids and list positions of their barycenters at the final time $T = 1.0$ in Table 7.

6.4. Two cylinders interacting with a shock. We simulate a more complicated problem with a shock interacting with two cylinders. Once again con-

TABLE 7
Positions of the barycenters of the oval disk and the rectangle at $T = 1.0$ as the mesh refines.

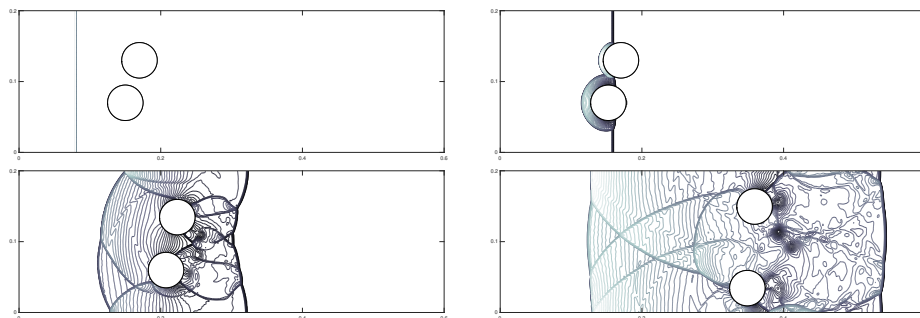| Cell number	400×80	600×120	800×160
Oval disk	$(0.6116, 0.1608)$	$(0.6077, 0.1607)$	$(0.6072, 0.1605)$
Rectangle	$(0.7718, 0.1397)$	$(0.7715, 0.1278)$	$(0.7720, 0.1211)$

FIG. 14. *The pressure profiles of the flow field and the positions of two cylinders at $T = 0$ (upper left), $T = 0.08$ (upper right), $T = 0.24$ (lower left), and $T = 0.48$ (lower right).*

TABLE 8
Positions of the barycenters of the two cylinders at $T = 0.48$ as the mesh is refined.

Cell number	150×50	300×100	450×150	600×200
Upper cylinder	$(0.3556, 0.1456)$	$(0.3579, 0.1485)$	$(0.3587, 0.1492)$	$(0.3591, 0.1496)$
Lower cylinder	$(0.3473, 0.0379)$	$(0.3475, 0.0358)$	$(0.3481, 0.0352)$	$(0.3489, 0.0344)$

sider a tunnel 0.2 units high whose top and bottom boundaries are both reflective walls. A Mach 3 shock initially locates at $x = 0.08$. The preshock fluid state is $(\rho_0, u_0, v_0, p_0) = (1.3, 0, 0, 0.1)$. The specific heat ratio of the gas is $\gamma = 1.4$. The two cylinders are positioned with barycenters at $(0.15, 0.07)$ and $(0.19, 0.13)$, respectively. The radius and the density of the cylinders are $r_\Omega = 2.5$ and $\rho_\Omega = 10\rho_0$. A computation is carried out in the region $[0, 0.6] \times [0, 0.2]$ using 600×200 cells. The pressure profiles of the flow field and positions of two cylinders at $T = 0$, $T = 0.08$, $T = 0.24$, and $T = 0.48$ are displayed in Figure 14. It can be seen that the two cylinders are pushed away by the higher pressure between them.

At the final time $T = 0.48$, the positions of the barycenters of the two cylinders obtained by an increasing resolution are listed in Table 8.

7. Discussions. In this paper we observe from the classical piston problem in gas dynamics that the acceleration is an important element in the description of moving boundaries, and we develop a one-sided GRP solver to design a high order moving boundary tracking algorithm, extending the MBT scheme originally developed in [14]. As analyzed, the newly developed MBT scheme is of second order and exhibits the desired performance.

As concluding remarks, we would like to make the following discussions:

(i) It is a natural choice to use the pair $(u_c, \frac{du_c}{dt})$, the velocity and the acceleration, to describe the motion of moving boundaries. In the literature, this observation is not directly integrated into the design of high order schemes, possibly due to the absence of the one-sided GRP solver. Of course, this solver can be equivalently derived as in the context of the gas kinetic scheme

[38] or approximated, e.g., in [37], with applications in other frameworks such as the adaptive grid method [28, 2] and the overlapping grid method [10, 32].

(ii) This method can be also extended to higher order versions, such as the fourth order version using the two-stage fourth order framework in [26, 25]. In addition to the time stepping procedure, several issues need to be settled. First, a high order spatial reconstruction is needed, such as the Hermite WENO reconstruction in [12]. Second, we need to approximate the complex geometry of rigid bodies with high order accuracy, which may be achieved through the non-uniform rational B-splines (NURBS) used in [21]. This is left for future work.

(iii) The present paper just studies the moving boundary tracking of solid bodies. A careful check of the methodology stated in section 2 indicates that this method can be extended to track various types of moving boundaries, e.g., in the context of multimaterial flows and viscous flows. The key ingredient is how to evaluate the force exerted on the moving surface to compute the acceleration.

(iv) Another meaningful investigation may be the extension of the MBT scheme to three-dimensional cases. A main difficulty lies the treatment of more complex geometry. As pointed out in [11], there are just 3 distinct cases for two-dimensional computations but 14 distinct cases for three-dimensional computations.

REFERENCES

[1] M. ARIENTI, P. HUNG, E. MORANO, AND J. E. SHEPHERD, *A level set approach to Eulerian–Lagrangian coupling*, J. Comput. Phys., 185 (2003), pp. 213–251.

[2] N. BARRAL, G. OLIVIER, AND F. ALAUZET, *Time-accurate anisotropic mesh adaptation for three-dimensional time-dependent problems with body-fitted moving geometries*, J. Comput. Phys., 331 (2017), pp. 157–187.

[3] T. J. BARTH, *Numerical methods for gasdynamic systems on unstructured meshes*, in An Introduction to Recent Developments in Theory and Numerics for Conservation Laws, D. Kröner, M. Ohlberger, and C. Rohde, eds., Springer-Verlag, Heidelberg, 1999, pp. 195–285.

[4] M. BEN-ARTZI AND J. FALCOVITZ, *Generalized Riemann Problems in Computational Fluid Dynamics*, Cambridge University Press, Cambridge, 2003.

[5] M. BEN-ARTZI AND J. LI, *Hyperbolic balance laws: Riemann invariants and the generalized Riemann problem*, Numer. Math., 106 (2007), pp. 369–425.

[6] M. BEN-ARTZI, J. LI, AND G. WARNECKE, *A direct Eulerian GRP scheme for compressible fluid flows*, J. Comput. Phys., 218 (2006), pp. 19–43.

[7] M. BERGER AND P. COLELLA, *Local adaptive mesh refinement for shock hydrodynamics*, J. Comput. Phys., 82 (1989), pp. 64–84.

[8] R. COURANT AND K. O. FRIEDRICHS, *Supersonic Flow and Shock Waves*, Springer, New York, 1948.

[9] G. DAKIN, B. DESPRÉS, AND S. JAOUEN, *Inverse Lax–Wendroff boundary treatment for compressible Lagrange-remap hydrodynamics on Cartesian grids*, J. Comput. Phys., 353 (2018), pp. 228–257.

[10] F. C. DOUGHERTY, J. A. BENEK, AND J. L. STEGER, *On Applications of Chimera Grid Schemes to Store Separation*, Tech. rep., National Aeronautics and Space Administration, Moffett Field, CA, 1985.

[11] J. DU, B. FIX, J. GLIMM, X. JIA, X. LI, Y. LI, AND L. WU, *A simple package for front tracking*, J. Comput. Phys., 213 (2006), pp. 613–628.

[12] Z. DU AND J. LI, *A Hermite WENO reconstruction for fourth order temporal accurate schemes based on the GRP solver for hyperbolic conservation laws*, J. Comput. Phys., 355 (2018), pp. 385–396.

[13] V. Dyadechko and M. Shashkov, *Reconstruction of multi-material interfaces from moment data*, J. Comput. Phys., 227 (2008), pp. 5361–5384.

[14] J. Falcovitz, G. Alfandary, and G. Hanoch, *A two-dimensional conservation laws scheme for compressible flow with moving boundaries*, J. Comput. Phys., 138 (1997), pp. 83–102.

[15] K. Feng and M. Qin, *Symplectic Geometric Algorithms for Hamiltonian Systems*, Springer, New York, 2010.

[16] H. Forrer and M. Berger, *Flow simulations on Cartesian grids involving complex moving geometries*, Internat. Ser. Numer. Math., 129 (1999), pp. 315–324.

[17] F. Gibou, R. Fediw, and S. Osher, *A review of level-set methods and some recent applications*, J. Comput. Phys., 353 (2018), pp. 82–109.

[18] I. Ginzburg and G. Wittum, *Two-phase flows on interface refined grids modeled with VOF, staggered finite volumes, and spline interpolants*, J. Comput. Phys., 166 (2001), pp. 302–335.

[19] J. Glimm, J. W. Grove, X. L. Li, and N. Zhao, *Simple front tracking*, Contemp. Math., 238 (1999), pp. 133–149.

[20] J. T. Grétarsson, N. Kwatra, and R. Fedkiw, *Numerically stable fluid-structure interactions between compressible flow and solid structures*, J. Comput. Phys., 230 (2011), pp. 3062–3084.

[21] N. Hosters, J. Helmig, A. Stavrev, M. Behr, and S. Elgeti, *Fluid–structure interaction with NURBS-based coupling*, Comput. Methods Appl. Mech. Engrg., 332 (2018), pp. 520–539.

[22] X. Y. Hu, B. C. Khoo, N. A. Adams, and F. L. Huang, *A conservative interface method for compressible flows*, J. Comput. Phys., 219 (2006), pp. 553–578.

[23] E. Lefrançois and J.-P. Boufflet, *An introduction to fluid-structure interaction: Application to the piston problem*, SIAM Rev., 52 (2010), pp. 747–767, https://doi.org/10.1137/090758313.

[24] X. Lei and J. Li, *Transversal effects of high order numerical schemes for compressible fluid flows*, App. Math. Mech. Engl. Ed., 40 (2019), pp. 343–354.

[25] J. Li, *Two-stage fourth order: Temporal-spatial coupling in computational fluid dynamics (CFD)*, Adv. Aerodynam., 1 (2019), 3.

[26] J. Li and Z. Du, *A two-stage fourth order time-accurate discretization for Lax–Wendroff type flow solvers* I. *Hyperbolic conservation laws*, SIAM J. Sci. Comput., 38 (2016), pp. A3046–A3069, https://doi.org/10.1137/15M1052512.

[27] L. Monasse, V. Daru, C. Mariotti, S. Piperno, and C. Tenaud, *A conservative coupling algorithm between a compressible flow and a rigid body using an embedded boundary method*, J. Comput. Phys., 231 (2012), pp. 2977–2994.

[28] M. Olim, H. Nagoya, K. Takayama, and F. Hiatt, *Entrainment of a spherical particle by the flow behind a shock wave*, in Shock Waves @ Marseille III, R. Brun and L. Z. Dumitrescu, eds., Springer, Berlin, Heidelberg, 1995, pp. 31–36.

[29] V. Pasquariello, G. Hammerl, F. Örley, S. Hickel, C. Danowski, A. Popp, W. A. Wall, and N. A. Adams, *A cut-cell finite volume – finite element coupling approach for fluid-structure interaction in compressible flow*, J. Comput. Phys., 307 (2016), pp. 670–695.

[30] C. S. Peskin, *The immersed boundary method*, Acta Numer., 11 (2002), pp. 479–517.

[31] M. A. Puscas and L. Monasse, *A three-dimensional conservative coupling method between an inviscid compressible flow and a moving rigid solid*, SIAM J. Sci. Comput., 37 (2015), pp. B884–B909, https://doi.org/10.1137/140962930.

[32] L. Qiu, W. Lu, and R. Fedkiw, *An adaptive discretization of compressible flow using a multitude of moving Cartesian grids*, J. Comput. Phys., 305 (2016), pp. 75–110.

[33] J. J. Quirk, *An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies*, Comput. & Fluids, 23 (1994), pp. 125–142.

[34] L. Schneiders, D. Hartmann, M. Meinke, and W. Schröder, *An accurate moving boundary formulation in cut-cell methods*, J. Comput. Phys., 235 (2013), pp. 786–809.

[35] K.-M. Shyue, *A moving-boundary tracking algorithm for inviscid compressible flow*, in Hyperbolic Problems: Theory, Numerics, Applications, S. Benzoni-Gavage and D. Serre, eds., Springer, Berlin, Heidelberg, 2008, pp. 989–996.

[36] S. Takeno, *Free piston problem for isentropic gas dynamics*, Japan J. Indust. Appl. Math., 12 (1995), pp. 163–194.

[37] S. Tan and C.-W. Shu, *A high order moving boundary treatment for compressible inviscid flows*, J. Comput. Phys., 230 (2011), pp. 6023–6036.

[38] K. Xu, *A gas-kinetic BGK scheme for the Navier-Stokes equations and its connection with artificial dissipation and Godunov method*, J. Comput. Phys., 171 (2001), pp. 289–335.