JOURNAL OF COMPUTATIONAL PHYSICS **138,** 83–102 (1997)
ARTICLE NO. CP975808

A Two-Dimensional Conservation Laws Scheme for Compressible Flows with Moving Boundaries

J. Falcovitz,* G. Alfandary,† and G. Hanoch†

*Institute of Mathematics, The Hebrew University, Jerusalem 91904, Israel; and
†Rafael Ballistic Center, P.O. Box 2250, Haifa, Israel
E-mail: *ccjf@math.huji.ac.il.

Received June 20, 1996; revised February 20, 1997

The generalized Riemann problem (GRP) scheme for the hydrodynamic conservation laws is extended to a two-dimensional moving boundary tracking (MBT) configuration, aimed at treating time-dependent compressible flows with moving (impermeable) boundary surfaces. A Strang-type operator splitting is employed for the integration of the conservation laws. The boundary motion is also split into Cartesian components that are taken with the respective operator-split integration phases. The conservation laws in boundary cells are integrated by a finite-volume scheme that accounts for changing mesh geometry (cell volume and side area through which mass, momentum, and energy fluxes are taken). The central feature of the scheme is the algorithms for evaluating the changing mesh geometry at boundary cells. These algorithms are based on a ''decomposition rule'' for evaluating polygon intersection area, which has been used extensively for rezoning in hydrocodes over the past decades. The decomposition rule is combined with the Cartesian splitting of the boundary motion, producing algorithms for the integration of conservation laws in boundary cells that are both consistent and simple. Consistency is taken to mean that the scheme produces an exact solution in the case of a uniform motion of the fluid and the boundary (common velocity). An illustrative example of shock lifting of a light-weight cylinder is presented.

1. INTRODUCTION

Time-dependent compressible flow phenomena involving shock waves and rarefaction waves [5] typically arise from sudden energy release or from rapid motion

0021-9991/97 $25.00

of a surface immersed in a compressible fluid. When the fluid is idealized as inviscid, the hydrodynamic phenomena are governed by the Euler equations expressing conservation of mass, momentum, and energy. These equations are routinely solved by resorting to shock-capturing finite-difference schemes, such as the GRP (generalized Riemann problem) scheme [2–4, 8].

In addition to a finite-difference scheme approximating the partial differential equations, a particular initial value problem requires a scheme for the initial data and the boundary conditions. In this presentation we consider flow problems that involve a particular class of boundary conditions: *a moving/deforming impermeable boundary surface immersed in the fluid and setting it in motion.*

Instances of such moving boundaries are the gas dynamics of reciprocating engines, flow through turbines or compressors having static and rotating blades, and more. In blast-safety analysis, the hurtling of a loose object (e.g., a vehicle) by an accidental blast wave may be treated as a compressible flow with moving boundaries. A simple example representing this class of phenomena is the shock lifting of a light-weight cylinder shown in Fig. 5 (Section 5 below).

Flows involving oblique, curved, or any ''non-Cartesian'' stationary boundaries are commonly treated by generating a body-fitted (i.e., ''boundary-fitted'') grid having a Cartesian image. As a typical application of such a grid, consider the study of double-wedge shock diffraction by Itoh *et al.* [11]. Recently, it was proposed by Olim *et al.* [14] to extend this boundary-fitting grid technique to the case of a moving boundary. The main idea is to repeat the grid-generation construction at small time increments as the boundary moves, adjusting the conservation laws scheme to account for the grid motion (by the so-called arbitrary Euler Lagrange, or ALE, extension). The disadvantage of this grid technique is that it is hampered by geometric limitations. In the case of the cylinder shock lifting (Fig. 5 below), which was also considered by Olim *et al.* [14], such geometric difficulties would arise if the cylinder is initially lying on the floor, or if two or more cylinders were to be considered. The treatment of moving boundaries by our proposed MBT (moving boundary tracking) method is free of such geometric limitations as outlined above, since it is based on a different grid concept.

Other approaches to the treatment of irregular boundaries embedded in a regular mesh have recently been reported, mostly for the case of a nonmoving (rigid) boundary. Evidently, our MBT scheme can also be used to treat a stationary irregular boundary, so that comparison with other schemes of this type is pertinent. Pember *et al.* [15] and Quirk [16] described a method for treating irregular stationary boundaries, where the accuracy and resolution in boundary cells were achieved by resorting to local mesh refinement. In these schemes a redistribution of conserved variables in boundary cells ensures stability without reducing the time steps to excessively small values. Wierse [20] employed an unstructured tetrahedra grid to simulate moving boundaries for modeling an internal combustion engine in three space dimensions. Bayyuk *et al.* [1] have employed a Quadtree-based Cartesian mesh refinement to treat inviscid flows with moving boundaries.

These schemes resort to a combination of local cell refinement and cell merging in order to integrate the conservation laws in boundary cells. In MBT, by contrast, cell merging is employed solely as an aid in treating excessively small cells as well

as newly born or newly covered cells. The central feature of the MBT scheme is the geometrically accurate treatment of the conservation laws in boundary cells, using an ordinary Cartesian grid.

While the formerly cited works employed an unsplit integration of the conservation laws and an adaptively refined mesh, a recently reported two-dimensional scheme by Forrer [9] uses split integration and an ordinary Cartesian mesh, employing an enlarged domain of dependence to ensure stability without excessively small time steps. This method seems closer to our MBT method, except for its reliance on local mirroring relative to the boundary line in order to obtain a stable integration, whereas in MBT small boundary cells are merged with contiguous larger cells for that purpose. Moreover, the MBT scheme maintains an exact integration of the conservation laws in boundary cells, whereas Forrer's scheme involves conservation errors in boundary cells [9].

Another approach to a moving boundary scheme is to employ a zone of body-fitted grid attached to the moving body, which is ''overlaid'' on the fixed computational mesh. This approach has been applied, for example, by Fujii [10], to the computation of flows with moving boundaries. In this case the scheme is quite different from our MBT and from the other previously mentioned schemes, and it constitutes a truly different alternate approach to the treatment of irregular boundaries.

The uniqueness of our MBT scheme lies in its property of consistent integration of the conservation laws in boundary cells and in the emphasis on relatively simple geometric algorithms that are employed to achieve that goal. Briefly stated, a scheme is consistent when it produces an exact solution in the case of a uniform flow (where both the irregular boundary and the fluid are moving at the same uniform velocity relative to the underlying mesh).

The principles of our MBT scheme are as follows. The boundary line is approximated by one or several polygons moving relative to an underlying Cartesian mesh (see Fig. 1). It is in effect a reduced-scope adaptation of the well-known coupled Euler–Lagrange (CEL) hydrodynamic interface scheme pioneered by Noh [13]. CEL is aimed at treating the motion of two immiscible dynamically interacting media. In one of the media the motion is treated in Euler coordinates, while in the other it is represented by Lagrange coordinates. Under the CEL scheme, the interface is a polygon consisting of free sides of Lagrange cells, its motion being determined in the same way as that of a free surface, but with the surface pressure taken as the pressure prevailing in the Euler cells intersected by the respective sides of the Lagrange interface. Thus, in the CEL scheme the coupled interface motion is determined by the Lagrange media, while the interface pressure is determined by the fluid in the Eulerian region.

Our present MBT scheme does not have this full coupling feature, as the motion of the boundary surface is assumed to be prescribed irrespectively of the flow which it induces. Just the same, some limited form of ''inertial coupling'' can be treated by the MBT scheme, as in the case of the cylinder shock-lifting example (Fig. 5). Here, the motion of a rigid body (i.e., the cylinder) is readily calculated from the total loading force due to the instantaneous pressure distribution on the moving boundary, so that the coupling is Euler–rigid body, rather than Euler–Lagrange as

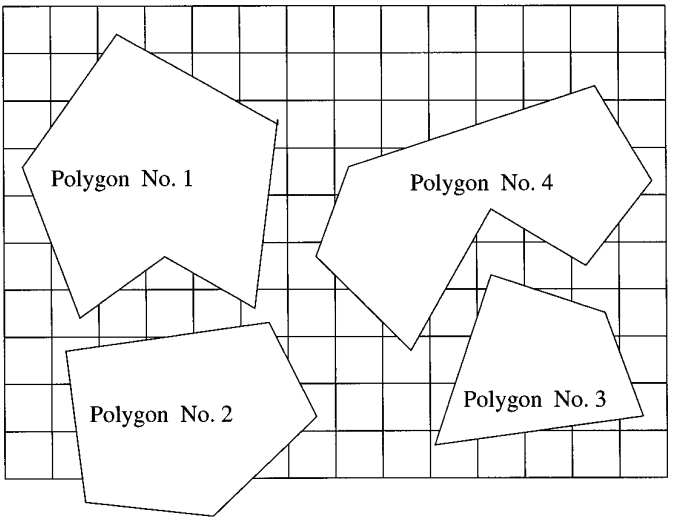

FIG. 1. Moving boundary tracking: multipolygon grid scheme.

in CEL. We employ the following key idea in order to enable a relatively simple computation of conservation laws in boundary cells. In spirit, this idea is an extension of Mason's [12] algorithm for evaluating the overlap area between two quadrilaterals by "decomposing" each quadrilateral into a union of the four trapezoids subtended by its sides. We thus refer to this methodology in the following as the *decomposition rule.*

The hydrodynamic conservation laws are integrated in regular and boundary cells by the GRP scheme [2–4, 8]. In fact, GRP is a one-dimensional conservation laws scheme, and the two-dimensional conservation laws are integrated by splitting them into a system of two one-dimensional conservation laws, which are then integrated via the GRP scheme.

It is well known that a flow in three-dimensional space can be analyzed in terms of two space coordinates, either when the flow field is plane-symmetric or when it possesses a cylindrical symmetry. In order to keep the presentation of the GRP/MBT scheme as simple as possible, we consider solely the plane-symmetric case, and we note that the extension of GRP/MBT to a cylindrical symmetry is fairly straightforward. However, since we prefer to keep in mind that either symmetry case is a reduced representation of a 3-D geometry, we usually retain the 3-D language in our 2-D geometric analysis. Thus, we shall refer to cell area as "cell volume," to side length as "side area," to a boundary segment length as "segment area," and to a boundary line as "boundary surface."

The plan of this paper is to start with an outline of the operator splitting and the one-dimensional GRP scheme in Section 2. When applied to the partial boundary cells formed by the moving boundary, this scheme takes a form which contains geometric parameters such as the exposed ("wet") cell volume and time-averaged exposed cell-interface area (side area). The resulting conservation laws scheme for boundary cells is presented in Section 3. The evaluation of the geometric parameters

V_i^n, V_i^{n+1} (cell volume), $A_{i+1/2}^{n+1/2}$ (time-averaged side area), and $Ab_i^{n+1/2}$ (time-averaged projected boundary area), all related to a cell i, involves some elaborate algorithms which are presented in Section 4. This part forms the central component of the MBT scheme, and most of the development effort of the scheme was devoted to it. As an illustration of the MBT scheme capabilities, a sample case of shock lifting of a cylinder is given in Section 5. This is followed by some concluding remarks in Section 6.

2. TWO-DIMENSIONAL CONSERVATION LAWS SCHEME

We provide a brief outline of the two-dimensional conservation laws scheme, which is based on operator splitting and the one-dimensional GRP scheme [2–4]. The Euler equations expressing conservation of mass, momentum, and energy in two space dimensions (and plane symmetry) for an inviscid compressible fluid are

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) + \partial_y \mathbf{G}(\mathbf{U}) = 0, \tag{1a}$$

$$\mathbf{U}(x, y, t) = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}; \quad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho E + p)u \end{bmatrix}; \quad \mathbf{G}(\mathbf{U}) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho E + p)v \end{bmatrix}; \tag{1b}$$

$$e = E - \frac{1}{2}(u^2 + v^2), \tag{1c}$$

$$p = (\gamma - 1)\rho e, \tag{1d}$$

where ρ, p, and (u, v) are the density, pressure, and (x, y) velocity components, respectively. E and e are the total and internal specific energies, respectively, and x, y, and t are the Cartesian coordinates and time. A perfect gas equation of state is assumed, where $\gamma > 1$ is the specific heats ratio.

The finite-difference approximation to (1a) is formulated as a Strang-type operator splitting [18], using the GRP scheme as the one-dimensional finite-difference operator. The unique feature of the splitting procedure is preservation of second-order accuracy; i.e., as GRP is second-order accurate, so is the two-dimensional conservation laws scheme using GRP as its one-dimensional "building block."

We begin by considering a splitting of the system (1a) into the following set of two simpler systems

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) = 0, \tag{2a}$$

$$\partial_t \mathbf{U} + \partial_y \mathbf{G}(\mathbf{U}) = 0. \tag{2b}$$

Loosely speaking, the system (2) is taken to mean that the evolution of an initial state $\mathbf{U}_0$ by (1a) over a short time interval Δt can be approximated by evolving $\mathbf{U}_0$ first subject to (2a) (over time Δt) obtaining a state $\mathbf{U}_1$, and then evolving $\mathbf{U}_1$ in accordance with (2b), again over time Δt.

Given a 1-D finite-difference approximation to (2a) and (2b), the operator-split (2-D) finite-difference approximation to (1a) is constructed as follows. Let $L_x(\Delta t)$, $L_y(\Delta t)$, and $L(\Delta t)$ denote finite-difference approximation operators for the integration by a time step Δt of (2a), (2b), and (1a), respectively. Strang [18] has shown that if $L_x(\Delta t)$ and $L_y(\Delta t)$ are second-order-accurate finite-difference approximation operators, then the operator sequence

$$L(\Delta t) = L_x\left(\frac{1}{2}\Delta t\right) L_y(\Delta t) L_x\left(\frac{1}{2}\Delta t\right) \tag{3}$$

is a second-order-accurate finite-difference approximation to (1a).

In a series of computations of shock wave phenomena, it was found out that the simplified sequence

$$\tilde{L}(\Delta t) = L_x(\Delta t) L_y(\Delta t) \tag{4}$$

provided results that were virtually indistinguishable from those obtained by (3). The following argument is an explanation of that fact. Consider the sequence obtained by repeating (3) a large number of times, i.e., for an integration of (1a) to time $T = N\Delta t$, where $N \gg 1$. Observe that the resulting N-fold operator sequence contains $N - 1$ pairs $L_x(\frac{1}{2}\Delta t) L_x(\frac{1}{2}\Delta t)$, each of which may be replaced by $L_x(\Delta t)$ to within a second-order accuracy. The resulting sequence is thus identical to an N-fold repetition of (4), in all but a single opening $L_x(\frac{1}{2}\Delta t)$ and a single closing $L_x(\frac{1}{2}\Delta t)$ operator. The error related to this difference may be neglected when $N \gg 1$. We thus normally use the splitting sequence (4) in all our computations, since it is simpler and more efficient than (3).

The foregoing heuristic arguments do not constitute a proof that the abbreviated splitting (4) is second-order accurate. We therefore retain the option to revert to the full Strang splitting (3), wherever such accuracy is essential. The MBT method can be employed with either splitting scheme, since its basic building block is a single-step split integration in either x-phase or y-phase. It is thus completely independent of the manner by which split integration phases are combined to result in a 2-D integration by one time step.

The one-dimensional operators $L_x(\Delta t)$ and $L_y(\Delta t)$ are given by the GRP scheme [2–4], which is briefly outlined as follows. The spatial domain is divided into equally spaced cells, so that cell i is the interval $x_{i-1/2} < x < x_{i+1/2}$, where $x_{i+1/2} = (i + 1/2)\Delta x$, $i = 1, 2, ..., i_{\max}$. The flow state $\mathbf{U}(x, y, t)$ is approximated at discrete time points $t_n = n\Delta t$ by a piecewise linear distribution in cells (for constant y), having the average value $\mathbf{U}_i^n$ in cell i. The finite-difference GRP conservation laws scheme, i.e., the operator involved in the relation $\mathbf{U}_i^{n+1} = L_x(\Delta t)\mathbf{U}_i^n$, is explicitly given by

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}\left[\mathbf{F}(\mathbf{U})_{i+1/2}^{n+1/2} - \mathbf{F}(\mathbf{U})_{i-1/2}^{n+1/2}\right], \tag{5}$$

where the time-centered fluxes $\mathbf{F}(\mathbf{U})_{i+1/2}^{n+1/2}$ are determined analytically from solutions to generalized Riemann problems that arise at cell interfaces $x_{i+1/2}$ as a result of

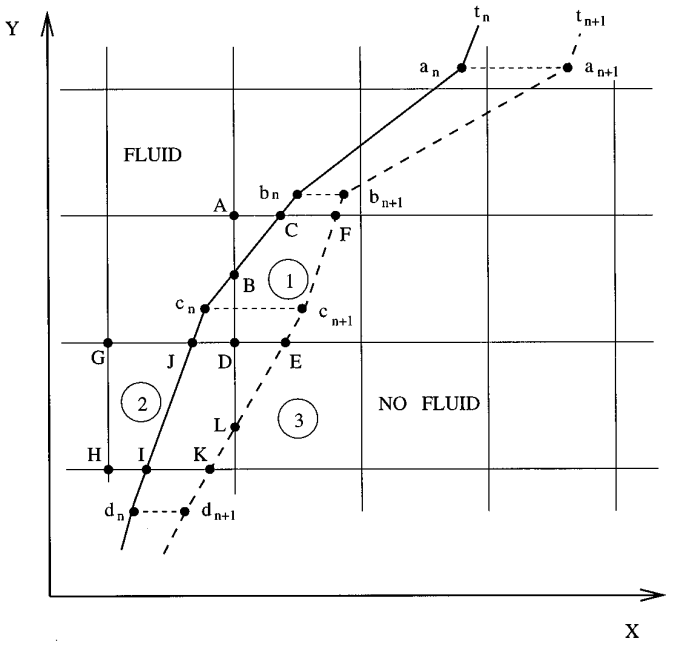

FIG. 2. *X*-split phase of moving boundary tracking.

the piecewise linear approximation to $\mathbf{U}(x, y, t_n)$ in cell i (belonging to some row of constant y). The unique feature of the GRP scheme is that those fluxes are evaluated from closed-form expressions obtained from an analytic solution to each generalized Riemann problem. We refer the reader to [2–4] for a comprehensive account of the GRP analysis and the resulting conservation laws scheme.

3. THE CONSERVATION LAWS SCHEME FOR BOUNDARY CELLS

Some geometric modifications are required in order to extend the conservation laws scheme (5) to boundary cells created by the intersection of a moving boundary with the underlying Cartesian mesh (see Fig. 2). A brief outline of this scheme was given in [7]. Here we provide a more comprehensive description of the scheme for the conservation laws of mass, momentum, and energy.

As mentioned in Section 1 above, the motion of the boundary nodes is split into (x, y) components, each taken at the respective operator-split integration phase (see Fig. 2 for boundary motion during the x-split phase). In the following we consider the x-split phase of the conservation laws, starting with the simplest one—the conservation of mass, where the density in boundary cell i obeys the following finite-difference relation

$$V_i^{n+1}\,\rho_i^{n+1} = V_i^n \rho_i^n - \Delta t[A_{i+1/2}^{n+1/2}\,(\rho u)_{i+1/2}^{n+1/2} - A_{i-1/2}^{n+1/2}(\rho u)_{i-1/2}^{n+1/2}]. \tag{6}$$

Here the time-level notation is as in (5), V_i^n is the "old" cell volume, V_i^{n+1} is the "new" cell volume, and the all-important "side area" $A_{i+1/2}^{n+1/2}$ is the exposed area

of cell interface $(i + 1/2)$ averaged over the time interval (t_n, t_{n+1}). The detailed algorithms for the evaluation of these parameters are the central part of the MBT scheme, and will be taken up in Section 4 below. In order to clarify the meaning of these parameters, we refer to the particular cell/boundary polygon configuration illustrated in Fig. 2. Consider cell 1, whose old image is the triangle ABC having volume V_i^n, and whose new image is the pentagon $ADEc_{n+1}F$ having volume V_i^{n+1}. As for the x-facing sides $(i \pm 1/2)$, $A_{i-1/2}^{n+1/2}$ is evaluated as an appropriately weighted average between the area of segment AB and the area of the fully exposed side AD (see details in Section 4 below). Since side $(i + 1/2)$ is covered throughout the time interval (t_n, t_{n+1}), its area is simply $A_{i+1/2}^{n+1/2} = 0$.

Several remarks concerning the mass conservation relation (6) are due. It is noted that (6) is in fact a "finite-volume" relation for the cell i taken as a time-changing control volume. We stipulate that in the case of uniform flow, where the boundary surface is an arbitrary polygon moving at the (uniform) fluid velocity, Eq. (6) would yield the exact solution, i.e., $\rho_i^{n+1} = \rho_i^n$. This stipulation leads to a particular way of defining the averaging procedure by which $A_{i+1/2}^{n+1/2}$ is evaluated (see Section 4 below). We also note that (6) does not contain mass flux through segments of the boundary surface. This is due to the assumption that the boundary surface is impermeable to the fluid. Thus, the advected parts of the momentum flux and the energy flux through the boundary surface must also vanish. However, momentum and energy fluxes through the boundary do not vanish identically. They correspond to boundary pressure and to work performed by the moving boundary, respectively, as we shall see below.

The x-component of the momentum flux obeys the following finite-volume relation in boundary cell i

$$V_i^{n+1}(\rho u)_i^{n+1} = V_i^n(\rho u)_i^n - \Delta t[A_{i+1/2}^{n+1/2}(p + \rho u^2)_{i+1/2}^{n+1/2} - A_{i-1/2}^{n+1/2}(p + \rho u^2)_{i-1/2}^{n+1/2}] - \Delta t A_{b_i} P_{b_i}, \tag{7}$$

where P_{b_i} is the boundary pressure averaged with respect to both the time interval (t_n, t_{n+1}) and the segment(s) of the boundary polygon contained in cell i. Just as Eq. (6) determines how $A_{i+1/2}^{n+1/2}$ is evaluated, Eqs. (7) and (9) indicate how A_{b_i} is determined as the time-averaged x-projection of the respective boundary segment(s). In fact, the above-mentioned averaging is performed for the *product* of boundary pressure and area. However, for the sake of simplified notation, it will be understood in the following that products of boundary pressure P_{b_i}, velocity component U_{b_i}, and area A_{b_i} are always averaged as the product, even though they are denoted as separate cell-related parameters. The sign convention for A_{b_i} is that it is positive when the x-component of the normal unit vector to the boundary segment pointing into the fluid is negative, and vice versa. The detailed evaluation of A_{b_i} is relegated to Section 4.3 below.

The boundary pressure P_{b_i} is evaluated for each boundary polygon side that intersects cell i within the time interval (t_n, t_{n+1}). It is determined by solving a "one-sided" Riemann problem (in fact, a "piston problem") in the direction normal to the boundary segment, where the normal component of the boundary velocity U^*

is known, and the result is the boundary pressure p^*. The initial data on the fluid side are taken as the pressure, density, and velocity component normal to the boundary segment in cell i at time t_n. In boundary cells we assume a uniform distribution of flow variables, as there is no meaningful way of assigning a piecewise linear distribution to the flow in irregularly shaped boundary cells. It is noted, therefore, that the MBT scheme reduces to first-order accuracy in boundary cells.

The y-component of the momentum is integrated by the relation

$$V_i^{n+1}(\rho v)_i^{n+1} = V_i^n(\rho v)_i^n - \Delta t[A_{i+1/2}^{n+1/2}(\rho uv)_{i+1/2}^{n+1/2} - A_{i-1/2}^{n+1/2}(\rho uv)_{i-1/2}^{n+1/2}], \tag{8}$$

where the boundary pressure flux term is absent (and will be accounted for at the y-split phase).

The finite-volume relation for energy conservation is

$$\begin{aligned} V_i^{n+1}(\rho E)_i^{n+1} = V_i^n(\rho E)_i^n &- \Delta t\{A_{i+1/2}^{n+1/2}[u(\rho E + p)]_{i+1/2}^{n+1/2} \\ &- A_{i-1/2}^{n+1/2}[u(\rho E + p)]_{i-1/2}^{n+1/2}\} \\ &- \Delta t A_{b_i} U_{b_i} P_{b_i}, \end{aligned} \tag{9}$$

where U_{b_i} denotes the x-component of the boundary velocity in cell i, and where the product-averaging procedure for the boundary work term is applied as mentioned above.

The sign convention mentioned above for A_{b_i} and the sign of U_{b_i} in the Cartesian coordinate system (x, y) yield the correct sign of the work term in (9).

An important assumption concerning the kinematics of the moving boundary surface relative to the underlying mesh is that fully exposed cells cannot become fully covered in one time step, or vice versa. This limitation is normally ensured by imposing the Courant–Friedrichs–Levy [17] stability constraint on the time step. With the magnitude of the time step limited in that way, to every partial cell that becomes fully covered or to a fully covered cell that becomes partially exposed, there will always be an adjacent exposed cell (i.e., a ''wet'' cell having a common side with the said cell). In such cases we treat that pair of cells as a single ''merged'' control volume, and the fluxes through their common side are immaterial and need not be evaluated. As an illustrative example, consider cells 2 and 3 in Fig. 2. Here cell 3 is newly exposed, so that the old control volume is the quadrilateral $IJGH$ and the new control volume is the ''merged'' quadrilateral $KEGH$ comprising the exposed parts of both cell 2 and cell 3. In this case, cell $IJGH$ does not require merging with the full cell to its left, since its length in the x-direction is about one-half of a full cell length. Only cells whose partial length is lower than a user-defined fraction (typically about 0.3) of a full cell length require such merging. It is noted that no ''unmerging'' procedure is ever needed. The flow variables evaluated for the finite volume of merged cells are simply assigned to each of the constituent (partial or full) cells.

In conclusion of this section, it is observed that the conservation relations (6)–(9) are readily interpreted as an application of the respective conservation laws to the boundary cells, which are treated as ''hybrid'' Lagrangian–Eulerian control vol-

umes. These rather simple relations are attained by lumping fine details of the geometry arising from the intersection of the moving boundary with the underlying mesh into the parameters V_i^n, V_i^{n+1}, $A_{i+1/2}^{n+1/2}$, and A_{b_i}. Of course, the operator splitting should also be credited with making this simplification possible. The detailed algorithms for the evaluation of these geometric parameters are presented in Section 4 below.

4. KINEMATICS OF BOUNDARY CELLS

Here we describe the procedures for evaluating the cell-related geometric parameters required in the MBT conservation laws scheme. The presentation of this topic is divided into three parts as follows. In Subsection 4.1, we specify the boundary configurations that are admissible by the MBT scheme, and then introduce the key idea of the "decomposition rule." Subsection 4.2 is devoted to the notation introduced in order to treat in concise language all relevant aspects of the intersection between a moving boundary polygon and the Cartesian mesh. Finally the algorithms for the evaluation of volume and area parameters in boundary cells are presented in Subsection 4.3.

Prior to presenting these, we state two general conventions regarding the notation in this section. First, it was found convenient to assume that all definitions and expressions pertain specifically to the y-split phase of the conservation laws (whereas in Section 3 above, the x-split phase was assumed). The respective x-split definitions and expressions can readily be derived by analogy. Second, we reiterate the following statement made in the introduction. Although two-dimensional plane symmetry is assumed here, a "3-D language" is retained by referring to a cell area as "cell volume," to a side length as "side area," to a boundary segment length as "segment area," and to a boundary line as "boundary surface."

4.1. *Boundary Scheme and Decomposition Rule*

The MBT scheme admits a moving boundary that in its most general configuration may consist of any number of nonintersecting closed curves that move and deform at a finite rate. By our convention, the fluid is always in the domain exterior to all boundary curves. Every boundary curve is assumed to be a closed simply connected curve that does not intersect itself, nor does it intersect any other boundary curve. It is noted, in particular, that boundary curves need not be convex, they may have discontinuous slopes (corner points) and they need not be wholly contained in the computational domain R_{xy}, which is the rectangle given by

$$R_{xy} := \{(x, y) \mid x_{\min} < x < x_{\max} \text{ and } y_{\min} < y < y_{\max}\}. \tag{10}$$

Every closed boundary curve is approximated by a closed polygon having an arbitrary number of vertices M. As an illustration of an admissible multipolygon boundary configuration, consider the example depicted in Fig. 1 above. Such configuration might be used, for instance, to treat several moving cylinders, as an extension of the single-cylinder shock-lifting example shown in Fig. 5 below.

As a result of the multipolygon capability of the MBT scheme, a particular cell may be intersected by two or more polygons. Hence, the evaluation of volume and side-area parameters pertaining to the exposed (i.e., ''wet'') part of that cell is best performed by evaluating the covered (i.e., ''dry'') cell volume and side area per boundary polygon. The multipolygon covered volume and side area are then obtained by adding the respective contributions of single polygons. We thus limit our discussion in the remainder of this section to a single polygon whose vertices are indexed $k = 1, 2, ..., M$, it being understood that the previously mentioned summation over multiple polygons will be performed as necessary.

We conclude this introduction by presenting what we believe to be the key idea for the evaluation of all cell-related and side-related geometric parameters. Consider the integral $\mathscr{A} = \oint -y dx$ which yields the area $\mathscr{A}$ surrounded by a closed line $\mathscr{L}$ in the (x, y) plane, where the integration along $\mathscr{L}$ is in the counterclockwise sense. Now, suppose that $\mathscr{L}$ is approximated by an M-sided polygon whose vertices are (x_k, y_k), $k = 1, 2, ..., M$. Then the polygon area A is given by

$$A = \sum_{k=1}^{M} A_k$$
$$A_k = \frac{1}{2}(y_k + y_{k+1})(x_k - x_{k+1}), \tag{11}$$

where A_k denotes the area of the trapezoid subtended by side $(k, k + 1)$ of the polygon. Clearly, the polygon area can be regarded as the union of all A_k trapezoids (with the appropriate $\pm$ sign of each trapezoid area). Hence, the intersection area of the polygon and a rectangular cell can also be expressed as the union (sum) of intersections between that cell and all trapezoids A_k.

In order to keep the algorithms based on such intersections simple, the entire trapezoid area must be of the same sign (positive or negative). We thus rule out the formation of a trapezoid whose two opposite sides intersect, i.e., a trapezoid having $y_k < 0$ and $y_{k+1} > 0$, or $y_k > 0$ and $y_{k+1} < 0$. (In fact, such trapezoid consists of two triangles, one having a positive area, the other having a negative area.) To comply with this stipulation, the point $y = 0$ is constrained to be outside the domain $(y_{\min}, y_{\max})$. (It is noted that in the case of cylindrical symmetry where y is the radial coordinate, this constraint is normally satisfied since it is assumed that $0 \leq y_{\min}$.)

When an intersection is expressed as the union of intersections with single trapezoids, we refer to it as a ''decomposed'' representation. It relies on the basic ''decomposition rule'' (11), where the polygon area is decomposed into the sum of trapezoid areas A_k. It is of interest to point out that this decomposition idea was originally proposed by Mason [12] for the evaluation of the intersection area of any two quadrilaterals, and has been extensively used in rezoning algorithms ever since. The advantage of a decomposition algorithm is its simplicity. In evaluating an intersection area using the decomposition rule, one obviates the (relatively) complex task of explicitly determining the vertices of the intersection polygon. This simplification is even more crucial to the evaluation of the time-averaged covered

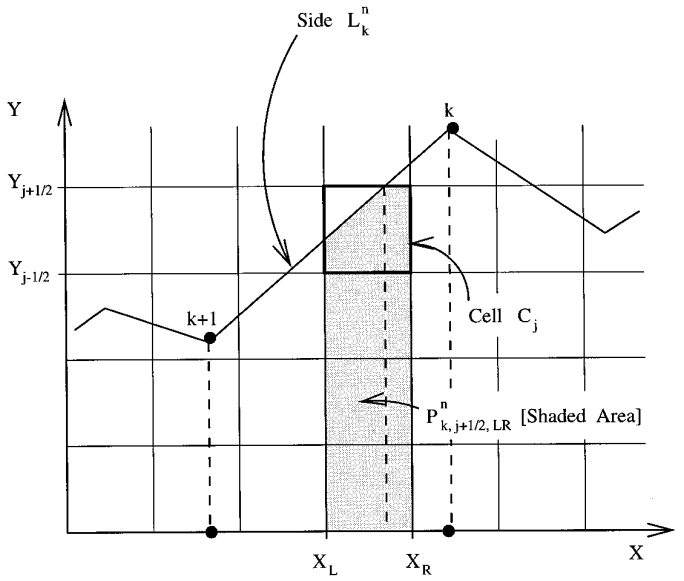

FIG. 3. Notations for polygon/grid intersection. Nodes of trapezoid T_k^n are indicated by ●.

area (length) of cell sides intersected by the moving boundary in the course of a single time step.

4.2. *Notation for Polygon–Mesh Intersection*

The elaborate task of evaluating volume and area parameters related to the intersection of a boundary polygon and the underlying mesh calls for adequate geometric definitions and notations, which is the subject of this subsection.

Considering the Cartesian mesh, it is obtained by dividing the computational domain R_{xy} (Eq. (10)) into rectangular cells of dimensions $(\Delta x, \Delta y)$. Thus, the grid of cell-interface points is given by

$$\begin{aligned} x_{i\pm1/2} &= x_{\min} + (i \pm 1/2)\Delta x - \Delta x/2, \quad i = 1, 2, ..., i_{\max}, \\ y_{j\pm1/2} &= y_{\min} + (j \pm 1/2)\Delta y - \Delta y/2, \quad j = 1, 2, ..., j_{\max}. \end{aligned} \tag{12}$$

Now, during the y-split phase, a single column of cells (constant x) is treated at a time. We shall consider in the following a fixed column denoted C_{LR}, whose bounds are $x_{\mathrm{L}} = x_{i-1/2}$ and $x_{\mathrm{R}} = x_{i+1/2}$, defined as

$$C_{\mathrm{LR}} := \{(x, y) \,|\, x_{\mathrm{L}} < x < x_{\mathrm{R}} \text{ and } y_{\min} < y < y_{\max}\}. \tag{13}$$

Within the column C_{LR}, we concentrate on a single cell C_j defined as

$$C_j := \{(x, y) \,|\, x_{\mathrm{L}} < x < x_{\mathrm{R}} \text{ and } y_{j-1/2} < y < y_{j+1/2}\}. \tag{14}$$

We refer to Fig. 3 for an illustration of column C_{LR} and cell C_j. Of particular use

is the rectangle denoted by $R_{j+1/2,\mathrm{LR}}$ subtended by side $y = y_{j+1/2}$ of C_j, which is defined as

$$R_{j+1/2,\mathrm{LR}} := \{(x, y) \mid x_{\mathrm{L}} < x < x_{\mathrm{R}} \text{ and } y_{\min} < y < y_{j+1/2}\}. \tag{15}$$

In regard to the moving boundary polygon, the following notation is adopted. Polygon vertices are denoted by (x_k, y_k^n) and (x_k, y_k^{n+1}), corresponding to the "old" time t_n and the "new time t_{n+1}, respectively. The vertices are indexed consecutively as $k = 1, 2, ..., M$ in a counterclockwise sense. It is noted that the time index has been omitted from x_k since during the y-split phase polygon vertices move in the y-direction and x_k remains constant. Individual polygon sides are denoted by L_k^n and L_k^{n+1}, corresponding to the straight segments joining vertices $(k, k + 1)$ at times t_n and t_{n+1}, respectively.

In the following, we denote by the name "trapezoid" any trapezoid in the (x, y) plane whose pair of parallel sides is lines of constant x. A special role is played by the trapezoid subtended by L_k^n or L_k^{n+1} and bounded by the line $y = 0$. Denoting these trapezoids by T_k^n and T_k^{n+1}, respectively, their nodes are

$$\begin{aligned} T_k^n &: (x_k, y_k^n), (x_{k+1}, y_{k+1}^n), (x_{k+1}, 0), (x_k, 0), \\ T_k^{n+1} &: (x_k, y_k^{n+1}), (x_{k+1}, y_{k+1}^{n+1}), (x_{k+1}, 0), (x_k, 0). \end{aligned} \tag{16}$$

In accordance with (11) above, the volume of T_k^n (denoted in (11) as the trapezoid area A_k) is given by

$$\mathrm{volume}(T_k^n) = \tfrac{1}{2}(y_k^n + y_{k+1}^n)(x_k - x_{k+1}), \tag{17}$$

where an analogous expression would be used for T_k^{n+1}. We reiterate the important fact that the trapezoid volume (17) may be positive or negative, and we refer to Fig. 3 for an example of a trapezoid T_k^n.

A further extension of L_k^n, T_k^n is needed in order to evaluate time-averaged covered side areas. We define a fractional time $0 < \theta < 1$ in the time interval (t_n, t_{n+1}), so that $L_k^{n+\theta}$, $T_k^{n+\theta}$ denote the corresponding interpolation between L_k^n, T_k^n and L_k^{n+1}, T_k^{n+1}, respectively. The exact definition of θ and the intermediate coordinates is

$$y_k^{n+\theta} = y_k^n + \theta(y_k^{n+1} - y_k^n) \tag{18a}$$

$$\theta = \frac{t - t_n}{t_{n+1} - t_n}. \tag{18b}$$

$L_k^{n+\theta}$ is then the polygon side $(k, k + 1)$ with $y_k^{n+\theta}$, $y_{k+1}^{n+\theta}$ as in (18a), and $T_k^{n+\theta}$ is defined as T_k^n in (16) with n replaced by $n + \theta$.

Two aspects of the geometry of intersection between the boundary polygon and the Cartesian mesh are considered here. The first is the "static" intersection, needed in order to determine the covered volume of boundary cells C_j at times t_n and t_{n+1}. To that end, we define the intersection polygon $P_{k,j+1/2,\mathrm{LR}}^n$ between a trapezoid T_k^n and a rectangle $R_{j+1/2,\mathrm{LR}}$ as

$$P^n_{k,j+1/2,\mathrm{LR}} := T^n_k \cap R_{j+1/2,\mathrm{LR}}. \tag{19}$$

It is readily verified that $P^n_{k,j+1/2,\mathrm{LR}}$ (if not a null set) is either a single trapezoid or the union of two trapezoids having a common side of constant x (see Fig. 3). In either case, its volume is easily evaluated, and its sign is identical to the sign of volume(T^n_k) given by (17), since the area of the rectangle $R_{j+1/2,\mathrm{LR}}$ is always taken as positive. The definition of $P^{n+1}_{k,j+1/2,\mathrm{LR}}$ is obtained by changing n into $n + 1$ in (19).

The second aspect of the polygon intersection with the mesh is a ''dynamic'' one, aimed at evaluating the covered area of y-facing sides of boundary cells C_j averaged over the time interval (t_n, t_{n+1}). For that purpose we define a ''polygon cover function'' $H(x, y, \theta)$ as

$$H(x,y,\theta) = \begin{cases} 1 & \text{point } (x,y) \text{ is inside boundary polygon at fractional time } \theta, \\ 0 & \text{point } (x,y) \text{ is outside boundary polygon at fractional time } \theta. \end{cases} \tag{20}$$

Clearly, the integration of $H(x, y, \theta)$ over the fractional time interval $0 < \theta < 1$ yields the ''cover fraction'' of any fixed point (x, y) during the time interval (t_n, t_{n+1}). In order to facilitate the computation of the time-averaged covered side area, this prescription is subsequently converted into a decomposed form by employing a ''trapezoid cover function'' $H_k(x, y, \theta)$ which is defined as

$$H_k(x,y,\theta) = \begin{cases} Z^{n+\theta}_k & \text{point } (x,y) \text{ is inside } T^{n+\theta}_k \text{ at fractional time } \theta, \\ 0 & \text{point } (x,y) \text{ is outside } T^{n+\theta}_k \text{ at fractional time } \theta, \end{cases} \tag{21}$$

where we have introduced the sign function $Z^{n+\theta}_k = \pm 1$ as the sign of the trapezoid $T^{n+\theta}_k$, defined by

$$Z^{n+\theta}_k = \mathrm{sign}[\mathrm{volume}(T^{n+\theta}_k)]. \tag{22}$$

Note that the sign function $Z^{n+\theta}_k$ is in fact independent of θ, since the assumption that the line $y = 0$ is not contained in R_{xy} (see 4.1 above) implies that the sign of the trapezoid $T^{n+\theta}_k$ is constant for $0 < \theta < 1$.

4.3. *Volume and Side Area in Boundary Cells*

The evaluation of exposed cell volume V^n_j, V^{n+1}_j for boundary cell C_j is readily performed by using the decomposition rule and noting the obvious relation

$$R_{j+1/2,\mathrm{LR}} = R_{j-1/2,\mathrm{LR}} \cup C_j. \tag{23}$$

Let the volume of $P^n_{k,j+1/2,\mathrm{LR}}$ be denoted by

$$Q^n_{k,j+1/2} = \mathrm{volume}(P^n_{k,j+1/2,\mathrm{LR}}); \tag{24}$$

then from (23) it follows that the covered volume $B^n_{k,j}$ of cell C_j is given by

$$B^n_{k,j} = Q^n_{k,j+1/2} - Q^n_{k,j-1/2}. \tag{25}$$

The covered volume B^n_j and the exposed volume V^n_j of cell C_j are then given by the decomposed expression

$$B^n_j = \sum_{k=1}^{M} B^n_{k,j} \tag{26a}$$

$$V^n_j = (x_R - x_L)\,(y_{j+1/2} - y_{j-1/2}) - B^n_j. \tag{26b}$$

The algorithm for V^n_j thus consists of the steps (24)–(26), where $Q^n_{k,j+1/2}$ are first evaluated for fixed k and all $j + 1/2$, then $B^n_{k,j}$ is evaluated by (25) for the same k and all j, and finally the summation in (26a) is updated by adding the contribution of $B^n_{k,j}$ to B^n_j for all j and a constant k. This procedure is repeated for each k for which side L^n_k intersects column C_{LR}. The new volume V^{n+1}_j is obtained by an analogous procedure.

We now take up the evaluation of the exposed side area $A^{n+1/2}_{j+1/2}$, which is considerably more elaborate than the evaluation of V^n_j, since an averaging over the time interval $t_n < t < t_{n+1}$ is required. As before, we seek the covered area $S^{n+1/2}_{j+1/2}$, so that the exposed area will be given by

$$A^{n+1/2}_{j+1/2} = (x_R - x_L) - S^{n+1/2}_{j+1/2}. \tag{27}$$

The basic guideline in establishing the algorithm for $S^{n+1/2}_{j+1/2}$ is that in the case of a uniform flow, where both fluid and boundary polygon move at the same (uniform) velocity, the finite-difference conservation laws (6)–(9) would yield the exact solution. The assumption of constant velocity during the time interval $t_n < t < t_{n+1}$ for each moving boundary node k, as implied by (18), is consistent with this requirement.

An averaging algorithm that meets the consistency requirement stated above is

$$S^{n+1/2}_{j+1/2} = \int_{x_L}^{x_R} dx \int_{t_n}^{t_{n+1}} d\theta\, H(x, y_{j+1/2}, \theta), \tag{28}$$

where the cover function $H(x, y, \theta)$ (see (20) above) serves, through the definite integral (28), to quantify the precise fraction of the time interval during which an infinitesimal side-area segment $(x, x + dx)$ is covered. Equation (28) results in an exact integration when both the fluid and the boundary move at some uniform velocity, since it is *formulated* that way.

While $S^{n+1/2}_{j+1/2}$ is completely defined by (28), given the coordinates of the polygon vertices during the time interval $t_n < t < t_{n+1}$ defined by (18), it is certainly not in a form suitable for computation. In order to recast (28) in a computationally feasible form, we resort to a decomposition rule for $H(x, y, \theta)$ based on the following lemma.

LEMMA 4.1. *The following decomposition rule holds for* $H(x, y, \theta)$,

$$H(x, y, \theta) = \sum_{k=1}^{M} H_k(x, y, \theta), \tag{29}$$

where $H(x, y, \theta)$ *is defined in* (20) *and* $H_k(x, y, \theta)$ *is defined in* (21).

Proof 4.1. Let an arbitrary point (x_0, y_0) within the computational domain R_{xy} be surrounded by a small simply connected closed polygon denoted by $\mathscr{P}_0$, whose area is A_0. Assume that $\mathscr{P}_0$ is traced in such a way that it is either wholly inside or wholly outside every trapezoid $T_k^{n+\theta}$ at $\theta = \theta_0$. Then, by the decomposition rule (11), and taking notice of the definition of $H_k(x, y, \theta)$ in (21), the area of the intersection of $\mathscr{P}_0$ and the boundary polygon at $\theta = \theta_0$ is given by $\sum_{k=1}^{M} H_k(x_0, y_0, \theta_0)A_0$. Noting the definition of $H(x, y, \theta)$ in (20), this intersection area is also equal to $H(x_0, y_0, \theta_0)A_0$. Upon division by A_0, the summation relation (29) is established.

Using the decomposition rule (29) for the point cover function, the time-averaged side area is decomposed into a summation of contributions $S_{k,j+1/2}^{n+1/2}$ due to polygon side L_k^n as

$$S_{j+1/2}^{n+1/2} = \sum_{k=1}^{M} S_{k,j+1/2}^{n+1/2}, \tag{30a}$$

$$S_{k,j+1/2}^{n+1/2} = \int_{x_L}^{x_R} dx \int_0^1 d\theta\, H_k(x, y_{j+1/2}, \theta). \tag{30b}$$

Generally speaking, the θ-integral in (30b), which we denote by the function $0 < \Theta_{k,j+1/2}(x) < 1$, is taken to mean the length of the fractional time interval during which point $(x, y_{j+1/2})$ is covered by trapezoid $T_k^{n+\theta}$ (with an appropriate $\pm$ sign). Let us assume that the motion of each polygon side $L_k^{n+\theta}$ is monotonic; i.e., its endpoints move at velocity $v_k^n \geq 0$, $v_{k+1}^n \geq 0$ (a justification of this assumption will be provided in the following). Then the time fraction cover function is defined as

$$\Theta_{k,j+1/2}(x) = \begin{cases} 0 & \text{point } (x, y_{j+1/2}) \text{ is outside } T_k^{n+\theta} \text{ for } 0 < \theta < 1, \\ Z_k^{n+\theta} & \text{point } (x, y_{j+1/2}) \text{ is inside } T_k^{n+\theta} \text{ for } 0 < \theta < 1, \\ Z_k^{n+\theta}\theta_c & \text{point } (x, y_{j+1/2}) \text{ is inside } T_k^{n+\theta} \text{ for } 0 < \theta < \theta_c \\ & \quad \text{and outside } T_k^{n+\theta} \text{ for } \theta_c < \theta < 1, \\ Z_k^{n+\theta}(1 - \theta_c) & \text{point } (x, y_{j+1/2}) \text{ is outside } T_k^{n+\theta} \text{ for } 0 < \theta < \theta_c \\ & \quad \text{and inside } T_k^{n+\theta} \text{ for } \theta_c < \theta < 1, \end{cases} \tag{31}$$

where by θ_c we denote the fractional time at which the status of point x on side $j + 1/2$ changes from covered to exposed or vice versa, as a result of intersection of this cell side with the moving polygon side $L_k^{n+\theta}$.

Adopting the definition of $\Theta_{k,j+1/2}(x)$ in (31), the average covered side area due to the moving polygon side $L_k^{n+\theta}$ is given by the finite integral

$$S_{k,j+1/2}^{n+1/2} = \int_{x_L}^{x_R} dx\, \Theta_{k,j+1/2}(x). \tag{32}$$

It is clear from the foregoing discussion that the function $\Theta_{k,j+1/2}(x)$ may readily be derived in closed form by considering the "sliding point" of intersection between the moving polygon side $L_k^{n+\theta}$ and side $j + 1/2$ of cell C_j. We illustrate this derivation

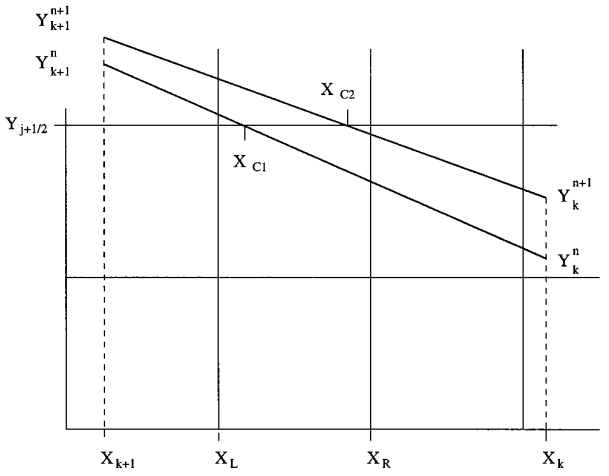

FIG. 4. Schematic intersection of moving boundary segment with side $j + 1/2$ of cell C_j for evaluating average covered side area.

by considering the particular instance shown in Fig. 4. In this case the intersection point slides from $x = Xc_1$ to $x = Xc_2$ as the fractional time increases from $\theta = 0$ to $\theta = 1$. After some algebra, the following expression is obtained for the time cover fraction in the range $Xc_1 < x < Xc_2$

$$\Theta_{k,j+1/2}(x) = \frac{(y_{k+1}^{n+1} - y_{j+1/2})(x - x_k) + (y_k^{n+1} - y_{j+1/2})(x_{k+1} - x)}{(y_{k+1}^{n+1} - y_{k+1}^n)(x - x_k) + (y_k^{n+1} - y_k^n)(x_{k+1} - x)}. \tag{33}$$

Since $\Theta_{k,j+1/2}(x)$ is a rational function, its definite integral in (32) is given in terms of elementary functions (note that in this case $\Theta_{k,j+1/2}(x)$ is constant for x outside the intersection range).

As previously mentioned, the algorithm for $S_{k,j+1/2}^{n+1/2}$ is constructed by assuming a "positive monotonic" motion of side $L_k^{n+\theta}$, i.e., $v_k^n \geq 0$ and $v_{k+1}^n \geq 0$, which leads to a significantly reduced programming complexity. There is no loss of generality entailed in this assumption, since any other case of motion where $v_k^n \leq 0$ and/or $v_{k+1}^n \leq 0$ can be transformed to the positive monotonic case, based on the following two arguments:

(i) The value of $S_{k,j+1/2}^{n+1/2}$ is unchanged if the motion of the polygon is reversed in time; i.e., the coordinate pairs y_k^n, y_k^{n+1} and y_{k+1}^n, y_{k+1}^{n+1} may be interchanged without affecting the value of $S_{k,j+1/2}^{n+1/2}$.

(ii) When the motion is nonmonotonic, i.e., $y_k^{n+1} > y_k^n$ and $y_{k+1}^{n+1} < y_{k+1}^n$, or vice versa, there exists a "pivot point" (x_p, y_p) on $L_k^{n+\theta}$ which remains stationary for $0 < \theta < 1$. Segment $L_k^{n+\theta}$ is then decomposed into the two segments $[(x_k, y_k^{n+\theta}), (x_p, y_p)]$ and $[(x_p, y_p), (x_{k+1}, y_{k+1}^{n+\theta})]$, each of which is treated separately as a monotonically moving segment.

Finally, it remains to specify the algorithm for evaluating the projected polygon area A_{b_j} per cell C_j, averaged over the time interval $0 < \theta < 1$. Since the difference between the area covered by $T_k^{n+\theta}$ on side $j + 1/2$ of C_j and the corresponding area

on side $j - 1/2$ is the projection of the segment of $L_k^{n+\theta}$ contained within C_j, the following definition of A_{b_j} is consistent with that geometric relation

$$A_{b_j} = \sum_{k=1}^{M} [S_{k,j+1/2}^{n+1/2} - S_{k,j-1/2}^{n+1/2}], \tag{34}$$

where it is readily verified that the sign convention for A_{b_j} (as given in Section 3 above) also holds. Note that in the multipolygon case, the summation in (34) is to be performed separately for each polygon, since A_{b_j} here is related to a single polygon. The terms containing A_{b_j} in the conservation laws (7) and (9) will then be extended to the multipolygon case by summation over the polygons that intersect each particular cell.

5. EXAMPLES

The only MBT test case for which a validating experiment exists so far is one where the boundary was stationary. It is a particular shock diffraction pattern by a double wedge, where the shock was regularly reflected by the first wedge, and where the second wedge surface was parallel to the incident shock front. An excellent agreement between computed and measured lines of constant density was obtained [6].

A moving boundary example, for which no experimental validation exists at present, is the liftoff of a rigid light-weight cylinder, initially resting on the floor, by a planar shock wave. This simple example is meant to represent more realistic (and more complex) scenarios, such as the hurtling of objects by an intense blast wave, or the tumbling of rigid stores abruptly released from a flying aircraft. The data for this sample case are cylinder radius $R = 5$, computational rectangle dimensions 100×20 (divided into 200×40 cells), initial shock front positioned at distance 8 from left boundary, cylinder center is initially at (15, 5), and initial (preshock) velocity, density, and pressure of (0, 0), 0.0013, and 10^{-6}, respectively. The fluid was taken as perfect gas with a specific heats ratio $\gamma = 1.40$. The cylinder was approximated by a regular polygon of 50 sides, and its density was assumed to be 0.01 (roughly eight times that of the gas). The incident shock wave Mach number was $M_s = 3.0$, and the boundary conditions on the left were set as inflow with the corresponding postshock flow parameters. The right side boundary conditions were nonreflecting, while the top and bottom had rigid wall boundary conditions. The value of the CFL coefficient was about 0.5.

The cylinder position and fluid velocity diagram at several points of the liftoff trajectory are shown in Fig. 5. Note that the cylinder initial lifting force is due to the asymmetric reflection of the incident shock wave by the cylinder at the floor side. Evidently, the reflected pressure is higher near the floor, causing a liftoff effect. Both cylinder trajectory and the reflected/transmitted shock waves are distinctly visible in the computed flow, and could conceivably be observed in a shock tube experiment.

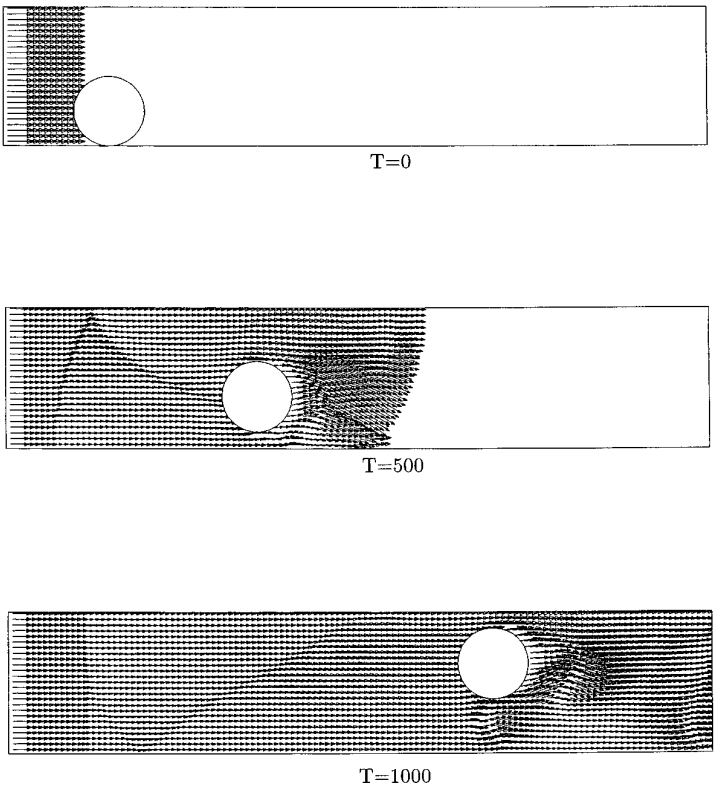

FIG. 5. Shock lifting of a light-weight cylinder. Shock Mach number $M_s = 3$. Cylinder/air density ratio = 8. Mesh size Rcyl/10.

6. SUMMARY AND CONCLUSIONS

The moving boundary tracking (MBT) method presented here significantly expands the scope of application of the operator-split conservation laws scheme GRP in two space dimensions. It was shown that the only modification needed in order to enable treatment of a moving boundary is the extension of the conservation laws finite-volume relations to a form containing time-dependent cell volumes and side areas. The evaluation of these geometric parameters is then taken as a separate task, which constitutes the bulk of the algorithmic and programming effort of the MBT method.

In accordance with the operator splitting of the conservation laws, the boundary nodes motion is split into (x, y) components that are taken with the respective split phase of the finite-difference integration. This boundary motion splitting has enabled simple formulation of the algorithms for evaluating geometric parameters, in particular the evaluation of the time-averaged exposed side area.

A central tool in the evaluation of the geometric parameters is the ''decomposition rule,'' which—in its plain form—is simply the classical relation giving the area of a polygon as a sum of the areas of the trapezoids subtended by its sides. Using the decomposition rule, all intersections of a moving polygon and the Cartesian mesh are

reduced to a ''decomposed'' form consisting of a summation of simpler intersections related to individual sides of the boundary polygon. An example of shock liftoff of a light-weight cylinder serves to illustrate a typical application of the MBT method.

REFERENCES

1. S. A. Bayyuk, K. G. Powell, and B. van Leer, *A Simulation Technique for 2-D Unsteady Inviscid Flows around Arbitrarily Moving and Deforming Bodies of Arbitrary Geometry*, AIAA Paper AIAA-93-3391-CP (1993).
2. M. Ben-Artzi and J. Falcovitz, A second-order Godunov-type scheme for compressible fluid dynamics, *J. Comput. Phys.* **55,** 1 (1984).
3. M. Ben-Artzi and J. Falcovitz, A high-resolution upwind scheme for quasi 1-D flows, in *Numerical Methods for the Euler Equations of Fluid Dynamics,* edited by F. Angrand, A. Dervieux, J. A. Desideri, and R. Glowinski (SIAM, Philadelphia, 1985), p. 66.
4. M. Ben-Artzi and J. Falcovitz, An upwind second-order scheme for compressible duct flows, *SIAM J. Sci. Statist. Comput.* **7,** 744 (1986).
5. R. Courant and K. O. Friedrichs, *Supersonic Flow and Shock Waves* (Springer-Verlag, New York, 1976).
6. J. Falcovitz, G. Alfandary, and G. Ben-Dor, Numerical simulation of the head-on reflection of a regular reflection, *Int. J. Numer. Methods Fluids* **17,** 1055 (1993).
7. J. Falcovitz, G. Alfandary, and G. Hanoch, Moving boundary tracking in compressible flow computation, in *Proceedings, JSME Spring Annual Meeting, Tokyo, Japan, March 1994.*
8. J. Falcovitz and M. Ben-Artzi, Recent developments of the GRP method, *JSME Int. J. Ser. B* **38,** 497 (1995).
9. H. Forrer, *Boundary Treatment for a Cartesian Grid Method,* Report 96-04, Seminar fuer Angewandte Mathematik, ETH Zuerich (1996).
10. K. Fujii, Unified zonal method based on fortified solution algorithm, *J. Comput. Phys.* **118,** 92 (1995).
11. K. Itoh, K. Takayama, and G. Ben-Dor, Numerical simulation of the reflection of a planar shock wave over double wedge, *Int. J. Numer. Methods Fluids* **13,** 1153 (1991).
12. D. S. Mason, Appendix in B. J. Thorne and D. B. Holdridge, *The TOOREZ Lagrangian Rezoning Code*, Sandia Nat. Lab. Rep. SLA-73-1057 (1974).
13. W. F. Noh, CEL: A time dependent two space-dimensional, coupled Eulerian–Lagrangian code, in *Methods in Computational Physics,* Vol. 3 (Academic Press, New York, 1964), p. 117.
14. M. Olim, H. Nagoya, K. Takayama, and F. Hiatt, Entrainment of a dust particle by a flow behind a reflected shock wave, presented at the *19th International Symposium on Shock Waves, Université de Provence, Marseilles, France, July 26–30* (1993).
15. R. B. Pember, J. B. Bell, P. Colella, W. Y. Crutchfield, and M. L. Welcome, An adaptive Cartesian grid method for unsteady compressible flow in irregular regions, *J. Comput. Phys.* **120,** 278 (1995).
16. J. J. Quirk, *Cartesian Grid Approach with Hierarchical Refinement for Compressible Flows: Final Report*, ICASE Report 94-51 or NASA CR-194938 (1994).
17. R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial Value Problems* (Interscience, New York, 1967).
18. G. Strang, On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* **5,** *506* (1968).
19. B. van Leer, Towards the ultimate conservative difference scheme V, *J. Comput. Phys.* **32,** 101 (1979).
20. M. Wierse, Solving the compressible Euler equations in time dependent geometries, in *Proceedings, 5th Int. Conf. on Hyperbolic Problems: Theory, Numerics, Applications, University at Stony Brook, New York, June 13–17, 1994,* edited by J. Glimm *et al.* (World Scientific, Singapore, 1994), p. 485.