MOCL — An Algorithm for Tracking Interfaces in Two-Dimensional Eulerian Codes*

Wen Wanzhi[1], Song Jiang[1], Yun Shourong[2]

[1]Laboratory of Computational Physics,
Institute of Applied Physics and Computational Mathematics,
P.O. Box 8009 (26), Beijing 100088, China
E-mail: wen_wanzhi@mail.iapcm.ac.cn

[2]Mechanical and Engineering Department,
Beijing Institute of Technology, Beijing 100081, China

We present an algorithm for tracking surfaces and material interfaces in two-dimensional Eulerian hydrocodes by combining the advantages of usual interface tracking methods with line segments. The main idea of the algorithm is that by defining the marker particles on cell boundaries, tracking a interface by traditional Lagrangian-type markers becomes tracking the interface by the markers at the Eulerian cell lines. We call this algorithm the Marker On Cell Line (MOCL) algorithm. In MOCL, the erosion and separation between materials are implemented by the transformation between material markers and the common markers of different materials, the sliding is approximated by using different transport velocities to different materials in a mixed cell. Moreover, the accuracy for computing material advection in the remap steps can be improved and the non-physical diffusion can be avoided by regarding the marker line in a mixed cell as the boundary of determining the fluid flux. On the other hand, some smearing effect is given by the MOCL algorithm, which can effectively tackle the numerical oscillations in mixed cells but brings only a negligible error for the interfaces. To test the algorithm, we incorporate the MOCL in a two-dimensional multi-material elastic-plastic Eulerian hydrocode (MHEP-MOCL) and present some numerical examples by the MHEP-MOCL code, including uniform motion of different objects, rotation of notched objects, jet formation from a liner and the penetration of a 10-mm-thick steel plate

* Supported by the Special Funds for Major State Basic Research Projects (No. G1999032801), the CNSF (No.10271019, the Jiechu grant) and the CAEP.

by the jet, and ogival and flat nose rods penetrating to concrete targets. Numerical results show that the algorithm is feasible and stable for computation of surfaces and material interfaces, in particular those materials with very complex surfaces.

1. INTRODUCTION

Many numerical methods for the solution of hydrodynamics problems fall into two basic categories: Lagrangian and Eulerian methods, based on the different coordinates. In a Lagrangian code, computational cells are generated to cover materials of interest and move with the materials, thus material interfaces can be well resolved and are automatically tracked in computation. However, under the motion of large deformation, the very fact that the computational cells exactly follow material particles can result in severe mesh deformation, causing the Lagrangian mesh distorted and even intersected each other, so that the computation becomes inaccurate and even breaks down. In general, Lagrangian codes perform well when applied to dynamic problems with small deformation. In the Eulerian description, the computational cells remain fixed in space and do not move with materials. Therefore, the transport of materials has to be calculated between cells in computation. One of the advantages of Eulerian methods is that the methods perform well for a number of dynamic problems with large deformation. On the other hand, for Eulerian methods it is difficult to capture high-accurately material surfaces and interfaces, in particular, in case that there are more than two materials in a computational cell. If the interfaces are not well-resolved at each time step, the non-physical diffusion would spread on the interfaces and results in the smearing of the interfaces. The smearing may form the zones of mixed fluids which would expand or penetrate into the zones of the pure materials, and consequently, the disastrous proliferation of mixed material zones and the spurious oscillation across the material boundaries occur sometimes, so that the computation breaks down. The above phenomenon can be observed in a number of numerical methods of approximating multi-material interfaces. Thus, for a Eulerian method one has to answer the crucial questions [1] such as how to determine high-accurately the position of material interfaces, how to compute physical quantities in mixed cells and how to compute the advection between mixed cells. If a material interface is resolved poorly, then the physical quantities which depend on the interface geometry (such as jet formation from a shaped charge liner) can not be computed correctly.

An interface is generally considered to be a surface across which one or more physical variables discontinue, and here we refer the interface to a material interface or a free surface. Three types of problems will arise in the numerical treatment of material interfaces [2]: (1) discrete representation of the interfaces, (2) evolution in time of the interfaces, and (3) the manner in which boundary conditions are imposed on the interfaces.

In this paper, we present a numerical method for tracking surfaces and material interfaces in two-dimensional Eulerian hydrocodes by carefully analyzing the above three types of problems and combining the advantages of usual interface capturing methods with line segments. The basic idea of the algorithm is that by defining the marker particles on cell boundaries, tracking a interface by traditional Lagrangian-type markers becomes tracking the interface by the markers at the Eulerian cell lines. We call this algorithm the Marker On Cell Line (MOCL) algorithm. This paper is organized as follows: In Section 2, we review some of the Eulerian interface capturing methods and discuss their advantages and disadvantages. Then, based on the discussion in Section 2, a new algorithm for tracking interfaces, Marker On Cell Line (MOCL), is proposed in detail in Section 3. In Section 4, the algorithm is briefly described in a two-dimensional multi-material elastic-plastic Eulerian hydrodynamics code (MHEP), in which the MOCL algorithm has been incorporated to track material interfaces. Numerical tests by the new code MHEP-MOCL are presented in Section 5. Numerical results demonstrate the high accuracy of the MOCL algorithm in the computation of highly complicated interfaces.

2. Related Interface Methods

There are varieties of numerical methods for treating material boundaries in Eulerian codes. In [2], they are divided into three different categories: the first is the function method; the second is the marker particle method; and the third is the line segment method. In this section, we will discuss the latern two methods and present more illustration and comparisons. The comparison of the advantages and disadvantages of these methods leads to a new method, the MOCL algorithm, which is flexible and efficient.

2.1. Marker Particle Methods

It is natural to track a interface with Lagrangian markers similar to Lagrangian coordinates, and the PIC (Particle In Cell) [3] and the MAC [4] methods are based on this idea. In both methods, the marker particles are spread over all material regions with each particle specified to move with the material velocity at its location. They are used to show the position of interfaces and follow the motion of the interfaces. But the PIC and MAC methods are different in nature. For PIC method materials are divided into a set of discrete particles with fixed mass, these particles accomplish the convective transport of mass, momentum and energy if they move from a cell to a neighboring cell. In contrast, the MAC method is to introduce a set of massless markers, and these markers do not involve in the numerical computation of flow fields unless in a mixed-fluid cell. The PIC method is quite effective for the numerical approximation of problems involving large distortion in two or three space dimensions.

Harlow and Amsden suggested an improved marker particle method [5], the GILA method, for which one does not need to spread marker particles over all material regions but only in several row cells from both sides of a interface, and the mass of the markers varies with the time step and is decided by the amount of the markers in a cell. So the transport is calculated in the continuous way when there is no marker in the cell, or in the discrete way when there are markers in the cell. A drawback of the method is that local numerical oscillation could be generated in the vicinity of interfaces.

In general, the main advantage of the marker particle methods is that it eliminates all logic problems associated with intersection of interfaces. Moreover, with the rapid development of high-speed and high performance computers, the drawback of the method requiring enormous computer storage and computation costs is getting negligible. On the other hand, the remaining difficulty is still the possible creation of artificial high or low marker densities in cells, due to the irregularity of flow fields, and this could result in numerical instabilities.

2.2. Line Segment Methods

The traditional techniques of approximating interfaces use marker points connected by line segments, resulting in a continuous line, or by short line segments in each cell, resulting in a discontinuous line. The approximations, therefore, are divided into two categories. The first is a tracking type, in which a set of massless tracer particles is initially positioned along the material boundaries and tracked by moving with the local materials. The second is a judging type, in which no tracer particles are used, but the volume function of each material in a mixed cell is used to construct the interface.

2.2.1. Tracking Type

Actually this type is developed from the MAC method. To apply the free surface stress condition at the correct free surface location within a Eulerian cell, Nichols and Hirt [6] initially specified a set of markers on an actual surface and move with the local material velocity. The surface is defined by the line segments connecting these markers in sequential order. Vander, Vorst and Rogers [7] modified the original MAC method by placing markers only on interfaces. Then, The interfaces are approximated by a chain of the markers connected by line segments and tracked by these markers moving at the local material velocity. This greatly saves computer storage. Similarly, in the HELP code [8], the massless marker particles were used to represent and track material interfaces, and the density of the markers can be increased or decreased in a specified region during the computation process. The successive line segment chains can not only track the interfaces, but also are used in the computation of certain physical and geometric quantities, and in particular, one utilizes the partial cell boundary areas determined by the interface position to compute the material flux between the Eulerian cells containing the interfaces. Unfortunately, the algorithm of treating material interfaces in the HELP code is too complicated logically and does not work well when the interfaces self-intersect.

2.2.2. Judging Type

For this type, tracking a material interface is indirect and Lagrangian-type marker particles are not adopted. The obtained interface is not continuous because the operation to find the interface in a mixed cell is lonely. To compute an interface at each time step, the mixes cells are first determined by judging whether a cell is mixed or by the value of the volume function of each material. Then, the more information of the interface in the mixed cells could be gained through its surrounding cells. Based on this idea, various techniques are developed. The 2DE [9] and MFLIC (multi-fluid in cell) [10] methods are proposed by representing roughly an interface only by the mixed cells. The methods are logically simple in (coding) but do not approximate the interface well. The SLIC [11] and VOF [2] schemes are to approximate an interface with straight lines in the cells, either perpendicular to or parallel to the coordinate direction in a calculated range; while the Youngs [12] and FLAIR [13] techniques approximate an interface by determining the slope of the straight lines in the mixed cells, and the interface position is used to compute the flux of each material across a cell boundary in the mixed cells. Generally speaking, the Youngs and FLAIR schemes are more accurate than the SLIC and VOF in the computation of an interface in a single cell but logically more complicated, and approximate interfaces by discontinuous lines. It should be pointed out that the volume function of each material in a mixed cell is the cornerstone of these methods.

Comparing with the tracking type, the judging type schemes save computer storage and computing time, and avoid complicated logical problems associated with marker particles. In the HELP code, employing the partial cell boundary areas occupied by the corresponding materials to compute the mass flux of materials in a mixed cell is rougher than strictly employing the sloped line segment such as the Youngs [12] and FLAIR [13] techniques, since the interface in the former is supposed to perpendicular to the cell boundary. However, the following problems may arise at an interface: (1) Non-physical diffusion of materials can be increased; (2) more and more mixed cells could be produced; (3) over-transport or under-transport in the remap steps could be induced. Hence, in order to construct a new algorithm which should approximate interfaces highly accurate, be logically simple and not be too diffusive, one has to circumvent these problems.

We should mention that Osher and Sethian in 1988 [14] proposed the level set method, a Eulerian method, to capture the evolution of surfaces which generalizes the height function method [2, 15] and is efficient, robust and simple. Since than, the level set method has been applied to many dynamic and image processing problems, and great progress has been made, see e.g. [16, 17], the monograph [18] and the review article [19], and among others.

3. Marker on Cell Line (MOCL) Algorithm

In this section, by absorbing some advantages of the line segment methods and by considering the fact that an actual interface is usually continuous, we present a new algorithm of capturing interfaces which is an extension of the line segment methods, logically simple and flexible. The basic idea [20] is that firstly, the material interface is initially defined through chains of line segments connecting the markers which are intersection points between the Eulerian rectangular mesh and each material boundary. Then, let the markers move for one time step Δt at the local material velocity to new position. The moved markers are called the transition markers, and the transition line is thus constructed by connecting the transition markers with line segments. The markers at new time step are obtained by finding out the intersection points between the transition line and the Eulerian rectangular mesh, and the interface at new time step is given by connecting the markers at new time step with line segments.

So, in this way the markers are always located on the cell boundaries. Secondly, the technique of sloped line segments in mixed cells is used to calculate strictly the flux of materials across the cell boundaries at the remap step. For our algorithm, tracking a material interface by traditional Lagrangian-type markers becomes tracking the interface by these markers on the Eulerian cell lines, we call it the Marker On Cell Line (MOCL) algorithm.

3.1. Defining The Material Markers

A set of massless marker particles are initially defined as the material markers, which are the intersection points between material interfaces and the Eulerian rectangle mesh. These markers are numbered in a counter-clockwise direction so that the materials are located on the left-hand side of the markers. The material interfaces, therefore, are defined by the line segments connecting the markers in the closed consecutive lines, i.e. the marker lines form the material packages (see Fig. 3.1). To prevent the boundary of the adjacent packages of different materials from crossing each other, we always call the markers on the common boundary the common markers. In Fig. 3.1, n_c stands for the total number of the common markers. The mixed cells are classed into the fully mixed cells in which there are common markers, and the no-fully mixed cells in which there is a void.

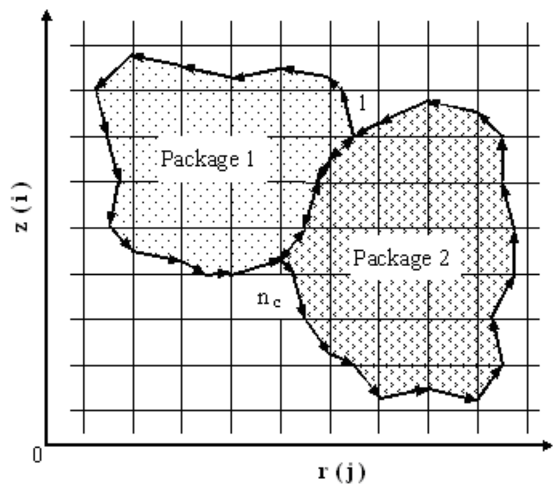


Fig. 3.1 Schematic of material packages in computation

In two space dimensions (r, z), the coordinates of the markers are given in grid line units, namely, letting a marker in a cell (i, j) where i corresponds to the z-direction and j to the r-direction throughout this paper (see Fig. 3.2), the coordinate of the marker is given by

$$M_c = k + C_c/h,$$

where M_c is the coordinate of the marker in the r- or z-direction, h is the spacing in the corresponding r- or z-direction, $k = j$ or $k = i$ is the number of the cell (i, j) in the corresponding r- or z-direction, C_c is the coordinate of the marker in a sub-coordinate system in the cell and is determined by the following regulation: (1) The markers on the left and bottom boundaries of the cell (i, j) belong to the cell; (2) the sub-coordinate system in the cell is defined by denoting the coordinate center and axes by the left-bottom node and the the left and bottom boundaries of the cell, respectively.

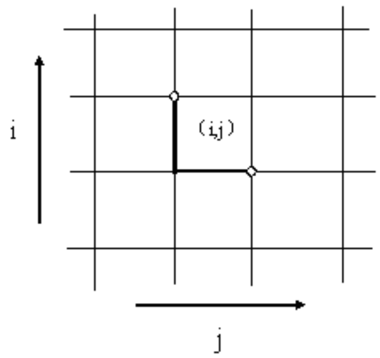


Fig. 3.2 Schematic of defining coordinates of a marker

3.2. Computing Time Evolution of the Markers

The evolution from time t^n to t^{n+1} of the markers use a two-step scheme. In the first step we compute the coordinate of the transition markers that are the markers at time t^n moved one time step Δt at the local material velocity, and in the second step we compute the coordinate of the new markers at time t^{n+1}.

3.2.1. Determining the coordinates of the transition markers

In this step the markers at time t^n move at the local material velocity to a new position after the time interval $\Delta t = t^{n+1} - t^n$, and hence the transition markers are obtained. The procedure is the following.

Firstly, compute the velocity components u and v at time t^n for a marker in r- and z-direction, respectively. The velocity components are obtained by an area overlap method which gives the area and the density weight to the velocity in the surrounding cells of the marker, see Fig. 3.3, i.e. we construct a rectangle with the center at the position of the marker and with the same size of the cell (i, j), which overlaps with the four cells $(i + 1, j - 1)$, $(i + 1, j)$, (i, j), $(i, j - 1)$ surrounding the marker. When a rectangle with the same size of the cell (i, j) is superimposed on the marker to be moved, then u and v are given by

$$u = \frac{\sum\limits_{L=1}^{4} \tilde{u}_L A_L \rho_L}{\sum\limits_{L=1}^{4} A_L \rho_L}, \quad v = \frac{\sum\limits_{L=1}^{4} \tilde{v}_L A_L \rho_L}{\sum\limits_{L=1}^{4} A_L \rho_L},$$

where L is the portion of the overlapped cells, A_L is the area of the overlapped part, $\tilde{u}_L$ and $\tilde{v}_L$ are the cell-centered velocity components of the cell corresponding to A_L in the r- and z-direction respectively, and ρ_L is the density of the material. The use of a density weight makes the marker move mainly with the heavier material, and avoids the interfaces unsteady across the strong density discontinuities.

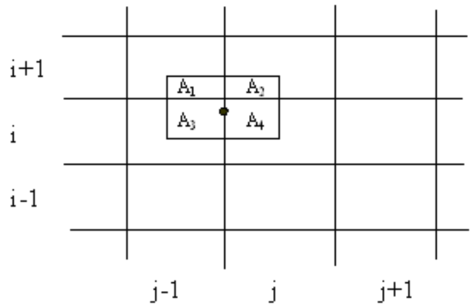


Fig. 3.3 Schematic of determining area and density weight for computing velocity components for marker evolution

Secondly, compute the coordinate of the transition markers. At the velocity (u, v) and after one time step Δt, the marker at time t^n moves to a new position and becomes a transition marker, the coordinate $(\tilde{r}, \tilde{z})$ of which is given obviously by

$$\begin{pmatrix} \tilde{r} \\ \tilde{z} \end{pmatrix}^{n+1} = \begin{pmatrix} r \\ z \end{pmatrix}^{n} + \Delta t \begin{pmatrix} u \\ v \end{pmatrix}^{n},$$

where (r, z) is the coordinate of the marker at t^n.

3.2.2. Determining the coordinate of the markers at time t^{n+1}

Fig. 3.4 shows the process of tracking a material interface from time t^n to time t^{n+1}. Firstly, the successive transition line at time t^{n+1} is gotten by connecting the transition markers by straight line segments. Secondly, the markers at time t^{n+1} are obtained by finding out the intersection points between the transition line and the Eulerian rectangular mesh. Finally, the new markers will replace the markers at t^n. Thus, at time t^{n+1} each material interface is given by connecting the new markers with line segments.

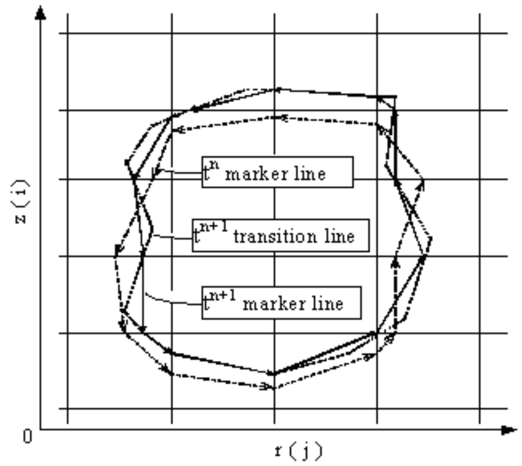


Fig. 3.4 Schematic of tracking material interface

3.3. Shape of the Material Boundary in a Cell

The geometry of the material boundary in a cell is formed only by marker lines and the geometry will be used in determining material flux across cell edges at the remap steps. In computation, however, the geometry of the material boundary could become so complicated that one can not deal with all possibilities in the computation of flux. Hence, in order to evaluate flux in MOCL, we consider only the following three cases in the way of the marker lines cutting the cell edges. All of the other cases can be reduced approximately to these three cases.

Case (1): Cells with two boundary markers on the cell edges

In this case, there are two shapes according to the relative position of two markers, in which the two markers (A and B) are on opposite sides in Fig. 3.5a or on adjacent sides in Fig. 3.5b.

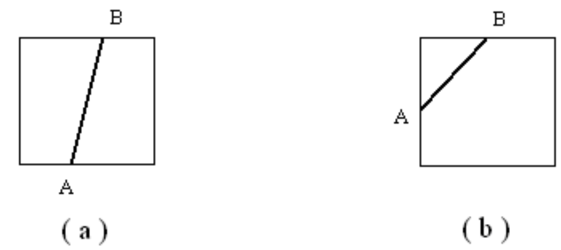


Fig. 3.5 Interface configurations for two markers case in a cell

Case (2): Cells with three boundary markers on the cell edges

In this case, only one shape is considered (see Fig. 3.6), in which the first and the third markers (A and C) are on opposite sides while the second marker (B) is on a adjacent side.

Case (3): Cells with four boundary markers on the cell edges

In this case, only one shape is also considered (see Fig. 3.7), in which the first and the fourth markers (A and D) are on opposite sides while the second and the third markers (B and C) are on a adjacent side.

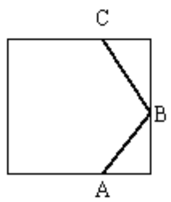


Fig. 3.6 Interface configuration for three markers case in a cell

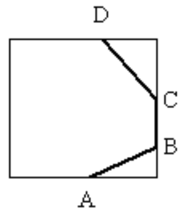


Fig. 3.7 Interface configuration for four markers case in a cell

3.4. Boundary Cell Problems

Suppose that there are two types of materials in consideration. Generally, a non-fully mixed cell can not be distinguished from a mixed cell and is regarded as fully mixed cell to deal with in many Eulerian codes, consequently, this will bring certain errors in the computation to mixed cells. In MOCL, a mixed cell is divided into a fully mixed cell and a non-fully mixed cell. In a fully mixed cell there are common markers and material interfaces between two materials, but in a non-fully mixed cell, these common markers and material interfaces are replaced with each material's own markers and free surfaces; the material with the bigger volume is denoted by the host-material and another by the servant-material. Therefore, a non-fully mixed cell means to introduce a void in the mixed cell and is approximated well in addition to the real boundary condition.

It should be noticed that there is small discrepancy between the volume fraction by a marker line and the actual volume fraction. Two situations maybe occur. The first is that although the volume fraction of a material by the marker line in a cell is a small value, the cell actually contains no material, and the second is that the marker line can cross a cell so that the cell is not contained in a material package but actually has a few materials. The principle treating this discrepancy is to fix the marker line which gives the location of an interface and the cell property. So, in the first situation, we fill the empty by a material with the volume that should be less than a prescribed upper bound (e.g., 0.05 times of the cell volume), and the physical quantities of the filled material are determined by the arithmetical mean method in the surrounding cells containing the material. While in the second situation, it is easy to take away the material from the cell. It should also be noticed that the free boundary condition is applied to non-fully mixed cells regardless of the host- or servant-material generally, however, if the sum of two volume fractions is equal to or greater than one, the pressure boundary condition should replace the free boundary condition in order to avoid the over-accelerated phenomenon.

3.5. Erosion and Separation between Material Packages

A closed marker line defines a type of material and represents a material package. The erosion and separation between the material packages are implemented by the transformation between material markers and the common markers of different materials. In this way the numerical simulation of impact and penetration processes can also be dealt with easily.

3.5.1. Marker intrusion between material packages

As two materials impact, the high pressure arises on the contact interface. Due to the free surface in the vicinity of the impact position, however, the pressure will decrease rapidly to the material strength. The basic idea of the numerical simulation of a impact process is that we first let the boundary markers of the two materials intrude into each other and then determine the common markers of the two materials to form the contact interface. The detailed algorithm (i.e. the determination of the common markers) is described in the following cases.

(1) In a non-fully mixed cell (i, j), there is a couple of markers (A,C) that intrude each other, and the marker A belongs to the material 1 and the marker C to the material 2 (see Fig. 3.8a). The intrusive line segment $|CA|$ is divided into two parts: CE and EA, and we have $|CE| = |CA|\, E_1/(E_1 + E_2)$, where E_1 and E_2 are the corresponding elastic moduls of two materials respectively. Thus a common marker E is defined.

(2) In a fully mixed cell (i, j), there are two couples of markers that intrude each other (see Fig. 3.8b). In this case, the two common markers E and F can be determined in the same manner as in (1).

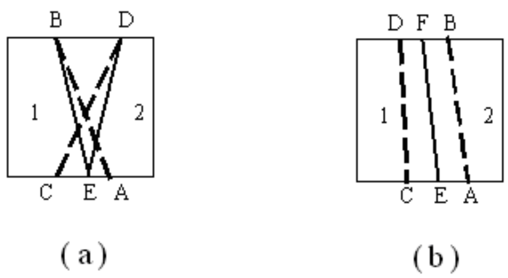


Fig.3.8 Schematic of marker intrusion and mergence

In the cases (1) and (2), only the linear elastic impacts are treated, where the intrusive line segments are divided into two parts that are inversely proportional to the materials' elastic moduls according to the pressure continuity condition. Similarly, the algorithm can be applied to the impact of other materials, such as a gas and a gas or a liquid and a liquid where the intrusive line segments are divided by their branch pressure, and also to the impact of two materials with different phases, such as a solid and a liquid or a solid and a gas where the intrusion is dealt with by either considering solid markers as the common markers or adjusting the materials' branch pressure.

3.5.2. Separation between material packages

The separation between material packages is implemented by changing the common makers to the individual material markers and then letting them move at different velocities. This technique is therefore applied only for the fully mixed cells. The criterion of the separation is whether the pressure in the fully mixed cells is less then zero or a specified small value (e.g. 10^{-6}). If a cell satisfies the separation condition, then the common marker on the cell boundary that is the closest to a free surface will be regarded as each individual material marker, and the substitute point of the common marker in each individual material is thus determined in the way described in the next paragraph (see Fig. 3.9). The substitute markers are employed to calculate the different velocities of the common marker by using the aforementioned area overlap method, such that the separation can actually occur. As a result, a void is introduced in the fully mixed cells and the fully mixed cells are changed into the non-fully mixed cells.

In Fig. 3.9, four cases are presented where the fully mixed cell (i, j) satisfies the separation condition and the dotted lines are the interfaces. In Fig. 3.9a, the common marker Q is located on the left half of the cell boundary AB. Hence, the points A and C are the substitute of the point Q to the materials 1 and 2 respectively, where $|QC| = |QB| + |AB|\,(\delta - 1)/2$. In Fig. 3.9b, the common marker Q is located on the right half of the cell boundary AB. Hence, the points C and B are the substitute of the point Q to the materials 1 and 2 respectively, where $|QC| = |QA| + |AB|\,(\delta - 1)/2$. In Fig. 3.9c, the common marker Q is located at one node of the cell (i, j). Hence, the points C and D are the substitute of the point Q to the materials 1 and 2 respectively, where $|QC| = |QA|\,(1 + \delta)/2$, $|QD| = |QB|\,(1 + \delta)/2$. For the sake of simplicity, we denote the first and last marker of the common markers in a cell by the input and output common markers of the cell, respectively. In Fig. 3.9d, the common marker Q is located between the input and output common markers of the cell (i, j). Hence, the points C and D are the substitute of the point Q to the materials 1 and 2 respectively, where the line segment CD is perpendicular to the cell boundary through the point Q, with $|QC| = |QD| = \Delta x(1 + \delta)/2$. In the above formulas $\delta \in (0, 1)$ is a parameter and decided in general by the space resolution and studied models.

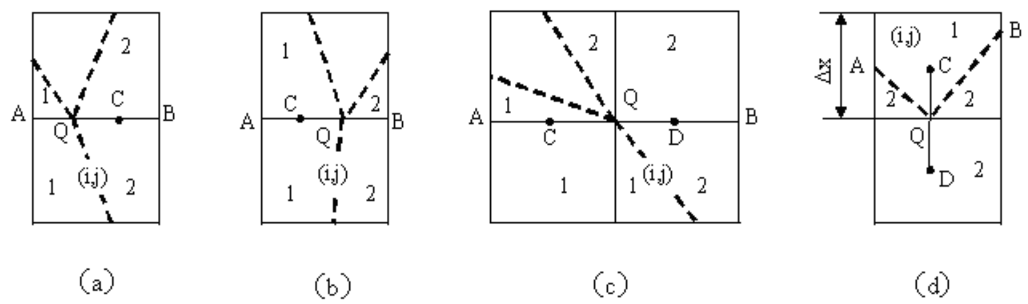


Fig.3.9 Schematic of common markers separation

3.6. Flux in Remap Steps

The remap step remaps the deformed cells to the original mesh. To this end, an appropriate amount of volume, mass, momentum, and energy must be transported between the cells. The r- resp. z-direction advection is calculated by utilizing the operator splitting technique. First, the transport velocity between cells is calculated. Then, the MOCL algorithm decides which materials in donor cells are moved, and then calculates the volume flux for a time step. Finally, the materials' mass, momentum and energy etc. are transported between the cells.

To consider the sliding between material interfaces, we use the following different algorithm to determine flux between the fully mixed cells.

3.6.1. Generally determining flux between cells

In this subsection we consider the case that excludes the determination of flux between fully mixed cells. In this case the transport velocity in one coordinate direction in a donor cell is uniform to each material.

Suppose that the cell (i, j) is a donor and the cell $(i, j+1)$ is an acceptor and the cell (i, j) is a free surface cell of the material 1, see Fig. 3.10a. The dotted line EF, which controls the maximal volume in the cell to be transported, is decided by the transport velocity and the time step. If the cell (i, j) does not contain any interface, then the line EF will determine the volume flux to the corresponding material. On the other hand, if the cell (i, j) contains a free surface, the line EF and the marker line together can determine the fluid flux, δV (i.e. the shadow in Fig. 3.10a). If the donor is a fully mixed cell, the limit line, EF, determines the general volume of flux δV, and the interface divides it into two parts (δV_1 and δV_2) for the two materials (1 and 2), as depicted in Fig. 3.10b. The corresponding mass flux is then

$$\delta m_1 = \rho_1 \delta V_1, \qquad \delta m_2 = \rho_2 \delta V_2,$$

where ρ_1, ρ_2 are the densities of the corresponding materials, respectively. The momentum and energy flux can be obtained similarly.

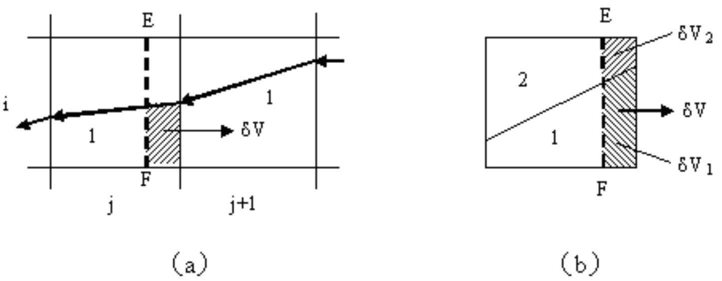


Fig. 3.10 Determination of material fluxes in a donor cell

3.6.2. Flux between fully mixed cells

The sliding phenomenon occurs at a material interface. The sharpness of the material interface is essential to the treatment of sliding. In MOCL, we utilize the idea that different materials in a mixed cell move at different velocities in a remap step [21], and the temporary velocity of each individual material in a fully mixed cell is obtained by linear interpolation between the fully mixed cell and one of its neighboring cells that contains no interface, so that with the help of the temporary velocity, the transport velocity is computed for each individual material. The algorithm is described as follows.

In Fig. 3.10, both of the cells (i, j) and $(i, j+1)$ are fully mixed, and the cells $(i+1, j)$ and $(i+1, j+1)$ contain only the material 1, and the cells $(i-1, j)$ and $(i-1, j+1)$ only the material 2. Suppose that the cell (i, j) is a donor and the cell $(i, j+1)$ is an acceptor, the temporary velocity $\tilde{u}_{1i,j}$ of the material 1 in the cell (i, j) in the r-direction (i.e. j-direction) is given by

$$\tilde{u}_{1i,j} = u_{i,j} + \frac{\Delta z_i (1 - \sigma_{1i,j})}{\Delta z_i + \Delta z_{i+1}} (u_{i+1,j} - u_{i,j}),$$

where $\sigma_{1i,j}$ is the volume fraction of the material 1 in the cell (i, j), Δz_i and Δz_{i+1} are the spacings of the corresponding column, $u_{i,j}$ and $u_{i+1,j}$ are the material velocities in the corresponding cells, respectively. Similarly, the temporary velocity of the material 1 in the cell $(i, j+1)$ is

$$\tilde{u}_{1i,j+1} = u_{i,j+1} + \frac{\Delta z_i (1 - \sigma_{1i,j+1})}{\Delta z_i + \Delta z_{i+1}} (u_{i+1,j+1} - u_{i,j+1}).$$

So, the transport velocity of the material 1 in the cell (i, j) in the r-direction is

$$\hat{u}_{1i,j} = \frac{\tilde{u}_{1i,j+1} \Delta r_j + \tilde{u}_{1i,j} \Delta r_{j+1}}{(\Delta r_j + \Delta r_{j+1}) + \Delta t (\tilde{u}_{1i,j+1} - \tilde{u}_{i,j})},$$

where Δr_j and Δr_{j+1} are the spacings of the corresponding column. In the same manner, the transport velocity $\hat{u}_{2i,j}$ of the material 2 in the cell (i, j) can be obtained. Finally, combining the interface line and the time step Δt, each volume flux (δV_1 or δV_2) can be evaluated as depicted in Fig. 3.11.

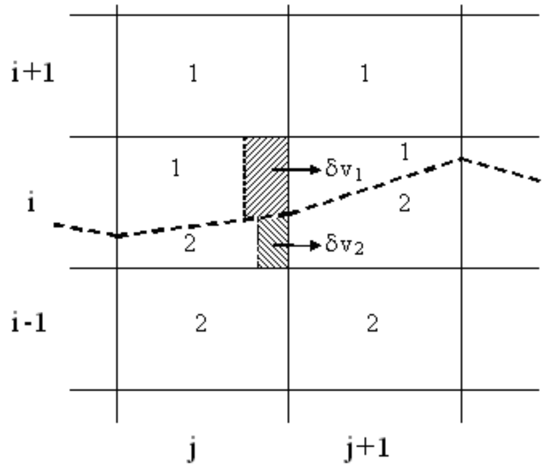


Fig. 3.11 Determination of material fluxes between full mixed cells

3.7. Approximation of Splits and Debris

In the process of highly dynamic events, splits, blocks and debris from fracture and spalling can often be produced on the target front and back. It is difficult to simulate numerically these phenomena. In MOCL, the splits from material edges are treated numerically by replacing markers, and the debris by forming the material subpackages.

See Fig. 3.12, suppose that the material package will split from the marker A, then the cell (i, j) (or $(i-1, j)$) has to satisfy the fracture condition of the material. In the failure cell the pressure and the deviator stress are assigned to zero, and the density is raised to the appropriate post-fracture value, so that a void is introduced in the cell. Thus the material in the cells (i, j) and $(i-1, j)$ is drawn out by that in their surrounding cells and becomes fewer and fewer. When the mass is less than a specified value, a new marker, i.e. the node B, will replace the marker A. As the process proceeds, the new marker C may replace the marker B. Finally, a split may occurs as the replacement process continues.

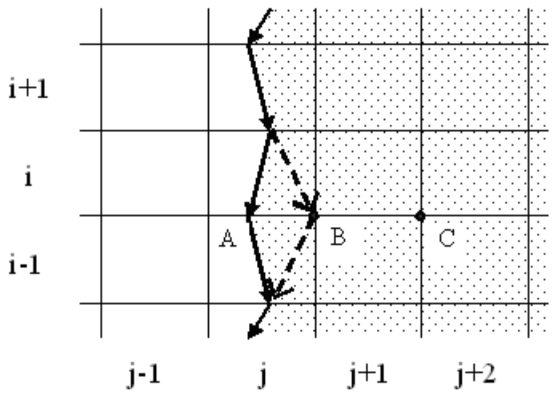


Fig. 3.12 Schematic of modeling split

In our algorithm the debris is generated by forming the material subpackages with the help of the material package deformation during the computation (see Fig. 3.13). A subpackage can be either removed immediately from the computation once it is generated, or tracked further. This depends on needs in practice.

By searching for a neck cell to a material package (cf. Fig. 3.14), the algorithm in this paper can judge whether a subpackage has been generated. A neck cell is referred to the cell, on the edges of which there are two input and two output markers of each individual material. In Fig. 3.14, on the boundary of the cell (i, j) the markers A and C are two input markers and the markers B and D are two output markers. To get a subpackage, we thus sum up the markers of two parts on both sides of the neck cell separately, and the part with the smaller value of the markers is regarded to as the subpackage. After taking away the subpackage, the markers of the new material package must be rearranged, where one input and one output markers in the neck cell are still left.

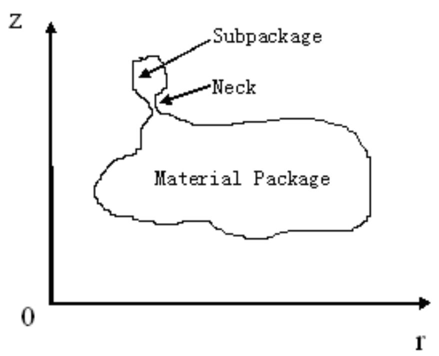


Fig. 3.13 Schematic of generating subpackage

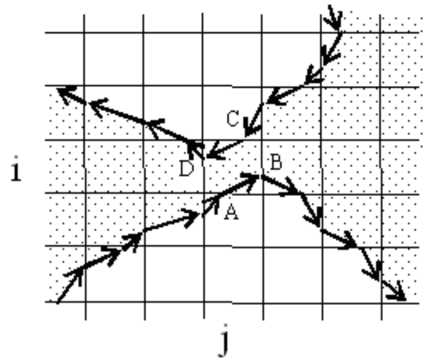


Fig. 3.14 Schematic of a neck cell

4. A Brief Description of The MHEP-MOCL Code

The MHEP-MOCL code [20] is an Eulerian hydrocode for modelling two-dimensional multi-materials with large deformation and strong shocks. It employs an Eulerian rectangular mesh and a first-order finite difference scheme to approximate the conservation laws of continuum mechanics. The space discretization is the central difference, while the time discretization is the forward difference. The pressure, the deviator stress, the density, the specific energy and the velocity components of the material are specified at the center of each Eulerian cell. The artificial viscosity for handling shocks is specified at the center of the upper and right boundaries of each cell. A Von Mises yield criterion with thermal softening, an instantaneous failure criterion and an accumulated damage model [22] are used, The numerical solution process is broken up into two steps: a Lagrangian step and a remap step.

The distinguishing feature of the MHEP-MOCL code is its capability to the physics which depends on geometry interfaces. The material interface is treated by the MOCL algorithm to mainly track the interface and determine the material flux. The numerical experiments in the next section show that the MOCL algorithm is efficient and flexible, and able to satisfy engineering needs.

5. Numerical Examples

In this section seven numerical examples are presented to demonstrate the accuracy and capabilities of the MHEP-MOCL code to solve numerically a variety of multi-material problems with large deformation problems. In all these examples, either experimental or analytical information is available for comparison with the numerical results.

5.1. Tests for Tracking Interfaces

Our algorithm for tracking material interfaces is effectively tested by whether a variety of objects with specific geometry will maintain their original shape under rotation or uniform motion. Five examples are tested here, where the objects are planar or axisymmetric and incompressible. The fourth example is the rotation of a notched disc (Zalesak's problem [23]), which was considered in [24], and the fifth example is the rotation of a concave plate based on the computation condition of the fourth. In our tests, t stands for time and am for the total mass of materials. It should be pointed out that the MOCL algorithm can regard those initial markers that play the key role in preservation of the material shapes as permanent markers and therefore, track interfaces accurately.

(1) One circle disc moving at a uniform velocity

The circle disc moves at a 45^0 angle with the z-axis across the computational meshed domain in two dimensions. The spacing steps are $\Delta z_i = \Delta r_j = 0.1cm$ uniformly, the diameter of the disc is $1.6cm$, the constant area density is $7.85g/cm^2$, and the constant velocity field is given by $u_z = u_r = 0.1mm/\mu s$. The numerical result at the time step $\Delta t = 0.0929\mu s$ is shown in Fig. 5.1, where the disc a is the initial shape with $am = 15.741g$, the disc b the shape at $t = 23.225\mu s$ with $am = 15.624g$ and the disc c the shape at $t = 46.450\mu s$ with $am = 15.332g$. Obviously, the shape of the disc scarcely varies, but its diameter is slightly reduced as its total mass decreases, and the relative error of its total mass is 0.026 at $t = 46.450\mu s$ in which the grid resolution is much low.

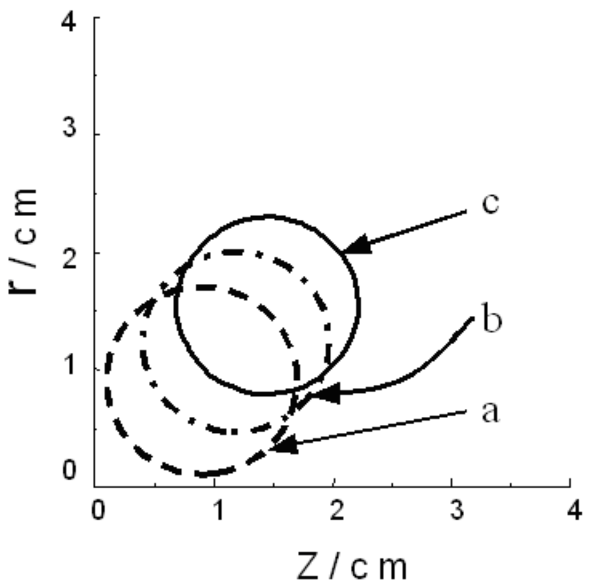


Fig. 5.1 Configurations of a moving disc

(2) One triangular plate moving at a uniform velocity

The plate moves at a 45^0 angle with the z-axis across the computational meshed domain in two dimensions. The space discretization and the initial condition for the plate are the same as in (1). The numerical result at the time step $\Delta t = 0.0929\mu s$ is shown in Fig. 5.2, where the plate a is the initial shape with $am = 15.700g$, the plate b the shape at $t = 23.225\mu s$ with $am = 15.668g$ and the plate c the shape at $t = 46.450\mu s$

with $am = 15.675g$. The result shows that the shape and size of the plate scarcely vary, and its total mass changes at random from $15.700g$ to $15.662g$.

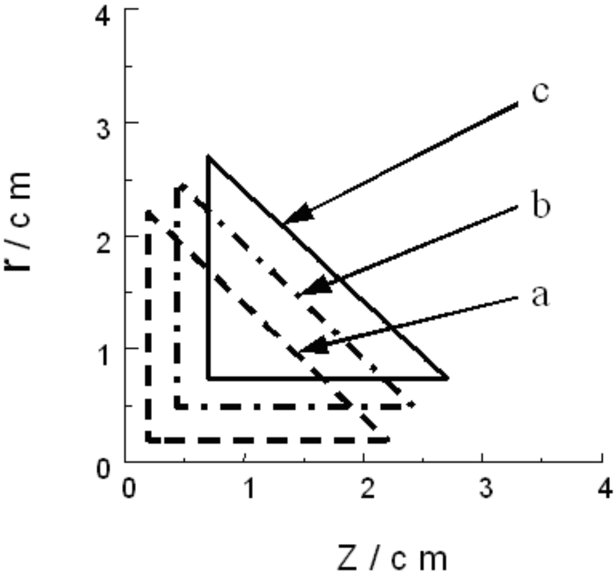


Fig. 5.2 Configurations of a moving plate

(3) One steel rod with ogival nose moving at a uniform velocity

The steel rod moves along the z-axis, which is the symmetric axis of the rod. The space discretization is the same as in (1), the diameter of the rod is $2cm$, the total length is $1.5cm$, the nose is initially given as parabolic shape, the constant volume density is $7.85g/cm^3$, and the constant velocity field is given by $u_{z0} = 0.2mm/\mu s, u_{r0} = 0mm/\mu s$. The numerical result at the time step $\Delta t = 0.0454\mu s$ is given in Fig. 5.3, where the rod a is the initial shape with $am = 92.530g$, the rod b the shape at $t = 22.704\mu s$ with $am = 92.530g$ and the rod c the shape at $t = 45.408\mu s$ with $am = 92.530g$. The result shows that the shape and the size of the rod scarcely vary, and its total mass is conserved.

(4) Rotation of a notched disc at a constant angular velocity

For this problem, we choose the computational condition as in [23] but give the change of the disc's area according to the grid dimensions. The computational grid is 100×100 cells in a $10cm \times 10cm$ periodic domain, the spacing steps are $\Delta z_i = \Delta r_j = 0.1cm$ uniformly. A notched circle of radius $1.5cm$ is positioned at cell $(50, 75)$, the width of the notch is $0.5cm$ and the height of the notch is $2.5cm$. The (stationary) velocity field is given by

$$u = \frac{\pi}{3.14}(5.0 - z), \quad v = \frac{\pi}{3.14}(r - 5.0).$$

In Fig. 5.4, we display the interface profile for the computation with a 100×100 grid. Obviously, comparing with the exact profile the computational arc (dotted line in

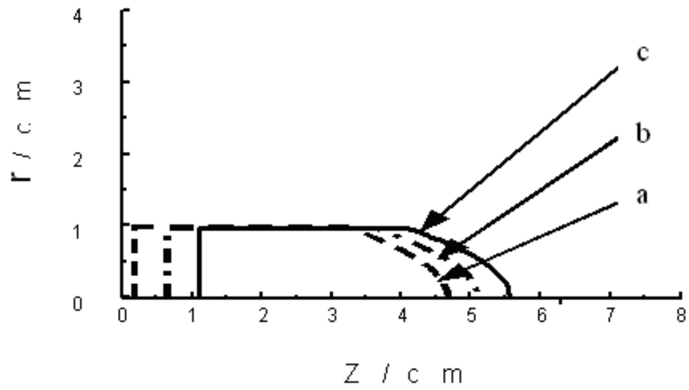


Fig. 5.3 Configurations of a moving rod

Fig.5.4) is slightly contracted and the relative error of its total area is 0.101. However, the straight lines are in almost exact agreement with the original configuration, and better resolved than in [24].

(5) Rotation of a concave plate at a constant angular velocity

In order to test the MOCL algorithm further, we choose the rotation of a concave plate. The space discretization and the initial condition for the plate are the same as in (4). A concave plate of dimension $1.2cm \times 1.2cm$ is positioned at cell $(50, 75)$, the width of the upper notch is $0.6cm$ and the height of the notch is $1.5cm$. In Fig. 5.5, the interface profile of the plate is displayed in its initial configuration and after one full revolution. Obviously the computational concave (dotted line in Fig.5.5) is in almost exact agreement with the initial profile, and its total area oscillates from $6.0900cm^2$ to $5.5649cm^2$. This also verifies that the MOCL algorithm performs much better for interfaces of straight line type than those of curve type.

5.2. Shaped Charge Calculation

A jet device is mainly composed of a copper liner with variable wall and a $\phi 40mm$ shaped charge [25]. The whole process includes four phases: high explosive detonation, liner collapse, jet formation and jet expansion. These phases are considered to be two-dimensional axisymmetric. The computation uses a nonuniform 220×100 grid. In Fig. 5.6, the initial configuration is shown for the two dimensional shaped charge problem. In Fig. 5.7, comparison with an X-ray of the experiment [25] at $t = 30\mu s$ shows that the relative error of the nose velocity of the jet is 0.116, the relative error of the length of the jet is 0.074, and the relative error of the diameter of the jet is 0.004, but the

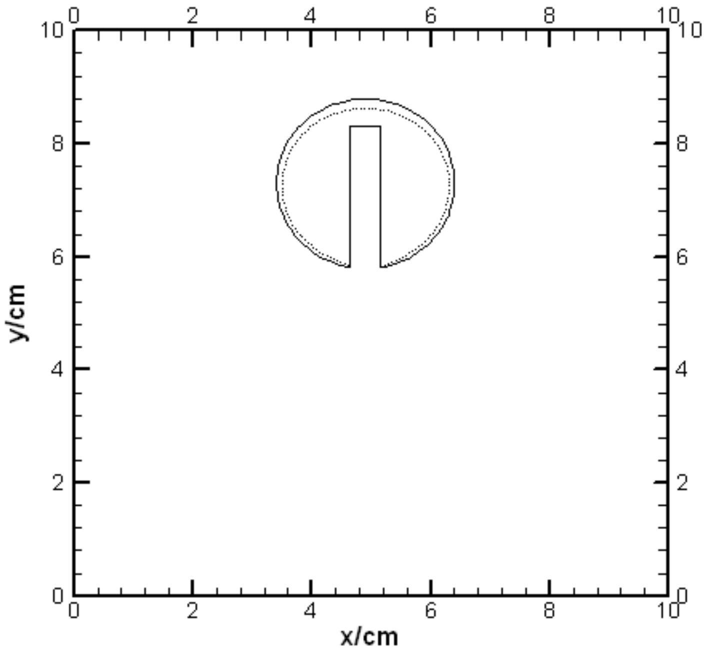


Fig. 5.4 Comparison with expected answer (dotted line) for Zalesak's problems after one full revolution of notched disc. Grid resolution 100×100, time is t = 62.8μs.

diameter of the slug is much bigger than the photograph and the slug not expanded out.

5.3. Penetration of a Steel Plate by a Jet

The simulation of the penetration of a steel plate by a jet is based on the shaped charge calculation in Subsection 5.2. In Ref. [25], a semi-infinite steel plate with thickness $10mm$ was located at the distance $60mm$ from the mouth of the liner. We take the computing data of the jet at $t = 21.615\mu s$ generated in Subsection 5.2 as the initial data of the jet here, when the distance from the jet nose to the mouth of the liner is $60mm$. The space discretization is the same as in Subsection 5.2, and the target, as material 2, is added in the computation according to Ref. [25]. The initial configuration is presented in Fig. 5.8. In Fig. 5.9, the outline of the penetration is demonstrated. Comparing with an X-ray of the experiment [25], the numerical result shows that the relative error of the residual velocity of the jet is 0.104, and the relative error of the aperture is 0.195. The two values are smaller than the experiment data because the impact velocity calculated in Subsection 5.2 is somewhat smaller than that of the experiment.

5.4. Effect of Shapes of Projectile Noses on Penetration

The shape of a projectile nose has effect on penetration. But numerical results can not illustrate this effect if interfaces in a hydrocode are not well tracked and resolved.

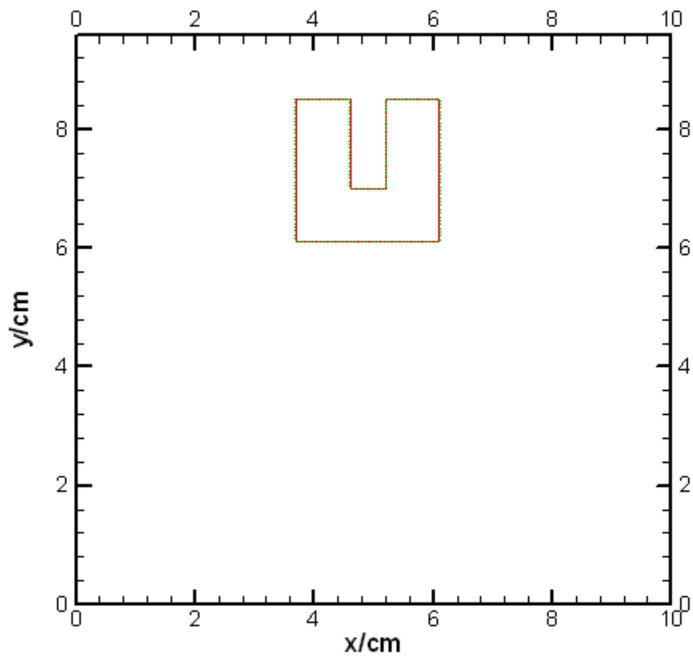


Fig. 5.5 Comparison of the initial interface profile for a concave plate and its profile (dotted line) after one full revolution. Grid resolution 100×100, time is t = 62.8μs.

In this test, the influence can be clearly observed, this demonstrates the efficiency of the MOCL algorithm. Here, the penetration into uniform semi-infinite concrete targets with ogival or flat nose projectiles is numerically simulated in the MHEP-MOCL code. The computation uses a nonuniform 250×85 grid. The initial configuration is presented in Fig. 5.10 and Fig. 5.12, respectively. From the numerical results, it is apparent to observe the effect of the shape of the projectile nose under similar circumstances. Firstly, the residual velocity with the ogival nose is greater than that with the flat nose. Secondly, the fracture range is remarkably different. The range with the ogival nose is more wider than that with the flat nose and its configuration is similar to alphabet W (see Fig. 5.11, where the curve $D = 1.0$ is a boundary of the complete failure.), while the configuration with the flat nose is obviously flatter and lower (see Fig. 5.13). And the overcharge with the ogival nose is far greater than that with the flat nose. These are in agreement with analytical results.

6. Conclusions and Suggestions

The tests in Section 5 show some major features on the MOCL algorithm:

(1) The MOCL algorithm is feasible and stable to treat material interfaces, especially those materials with very complex surfaces.

(2) The MOCL algorithm can improve accuracy for computing material advection in the remap steps in which the marker line in a mixed cell is used to strictly calculate the fluid flux across the cell boundary, and this avoids the non-physical diffusion in the

vicinity of material interfaces, consequently the computed interfaces approximate the real interfaces more accurately.

(3) The MOCL algorithm average a local interface in cells. Consequently, it can effectively eliminate the numerical oscillations in mixed cells. The MOCL technique makes a material surface slightly contract when it is convex, and slightly expand when it is concave. The related error, however, is very small, and much smaller if objects move along the coordinate directions.

(4) The erosion and separation between material packages are implemented by the transformation between the material markers and the related common markers. This algorithm makes the numerical simulation of impact and penetration processes quite easy. The numerical tests show that the algorithm is both efficient and practical.

(5) The sliding is approximated by employing different transport velocities to different materials in a mixed cell in the remap steps. The numerical tests show that in particular, the algorithm can hold back over-transport or under-transport and decrease the discrepancy between the volume fraction given by the marker line and the actual volume fraction.

All numerical examples in Section 5 are carried out for the fixed meshes. The accuracy and resolution of MOCL in the approximation of material interfaces will be improved when the adaptive mesh refinement technique is applied only in the vicinity of material interfaces. In many impact problems, cracks and hollows will be generated and evolute with time in the material interior. To simulate such problems, empty subpackages may be introduced to the interior destroyed cells and then tracked. The normal and tangential stress conditions may be applied at the location of interfaces to improve further the accuracy of the numerical simulation of sliding problems.

References

[1] D. Y. Li, G. R. Xu, H. S. Shui, G. Y. He, G. N. Chen and G. X. Yuan, Numerial Methods for Two-Dimensional Time-Dependent Fluid Dynamics, Science Press, China, 1987, p.7. (Chinese)

[2] C. W. Hirt and B. D. Nichols, Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries, *J. Comput. Phys.* **39,** 201 (1981).

[3] A. A. Amsden, The Particle-in Cell Method for the Calculations of the Dynamics of Compressible Fluids, Los Alamos Scientific Laboratory, LA-3466, 1966.

[4] J. E. Welch, F. H. Harlow, J. P. Shannon and B. J. Daly, the MAC MethodA Computing Technique for Solving Viscous, Incompressible, Transient Fluid Flow Problems Involving Free Surface, Los Alamos Scientific Laboratory, LA-3425, 1966.

[5] F. H. Harlow and A. A. Amsden, Multifluid Flow Calculations at all Mach Numbers, *J. Comput. Phys.* **16,** 1(1974).

[6] B. D. Nichols and C. W. Hirt, Improved Free Surface Boundary Conditions for Numerical Incompressible-Flow Calculations, *J. Comput. Phys.* **8,** 434(1971).

[7] M. J. Vander, Vorst and J. C. W. Rogers, Calculation of Vertical Water Entry of A Cone by the Partial Cell Marker and Cell Method, Proceedings of the 1976 Heat Transter and Fluid Mechanics Institute, Edited by A. A. Mckillop, J. W. Baughn and H. A. Dwyer, Davis, California, 1976, P.405.

[8] L. J. Hegeman and J. M. Walsh, HELP, A Multi-material Eulerian Program for Compressible Fluid and Elastic-Plastic Flow in Two Space Dimensions and Time, Report AD-726459, 1976.

[9] J. D. Kershner and C. L. Mader, 2DE: A two-dimensional Continuous Eulerian Hydrodynamic Code for Computing Multi-component Reactive Hydrodynamic Problems, Los Alamos Scientific Lab., Report LA-4846, 1972.

[10] G. R. Xu, Z. L. Yu, Z. M. Liao, X. C. Yuan, S. R. Zhou, An Eulerian Numerical Method for the Multi-Material Compressible Fluid Dynamics, *Jounal on Numerical Methods and Computer Applications,* **1,** 163(1980). (Chinese)

[11] W. F. Noh and Paul Woodward, SLIC (Simple Line Interface Calculation), Proceeding of the fifth International Conference on Numerical Methods in Fluid Dynamics, Springer Lecture Notes in Physics 59, 1976, P.330-340.

[12] D. L. Youngs, Time-Dependent Multi-Material Fluid Dynamics, Edited by K. W. Morton and M. J. Baines, *Academic Press,* 1982, p.273-285.

[13] N. Ashgriz and J. Y. Poo, FLAIR: Flux Line-segment Model for Advection and Interface Reconstruction, *J. Comput. Phys.* **93,** 449(1991).

[14] S. Osher and J. A. Sethian, Fronts Propagating with Currature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations, *J. Comput. Phys.* **79,** 12(1988).

[15] C. W. Hirt, B. D. Nichols and N. C. Romero, SOLA-A Numerical Solution Algorithm for Transient Fluid Flows, Los Alamos Scientific Lab., LA-5852, 1975.

[16] W. Mulder, S. Osher and J. A. Sethian, Computing interface motion in compressible gas dynamics, *J. Comput. Phys.* **100,** 209(1992).

[17] D. Peng, B. Merriman, S. Osher, H. Zhao and M. Kang, A PDE-Based Fast Local Level Set Method, *J. Comput. Phys.* **155,** 410(1999).

[18] J. A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge Univ. Press, 1999.

[19] S. Osher and R. P. Fedkiw: Level set methods: an overview and some recent results, *J. Comput. Phys.* **169**, 463(2001).

[20] W. Z. Wen, Marker on cell line (MOCL) and its application in numerical simulation, Ph.D dissertation, (Beijing Institute of Technology, China, 1998). (Chinese)

[21] S. Z. Feng, An Algorithm Study and Numerical Computation on Penetration and Damage of Concrete, Ph.D dissertation, (Beijing Institute of Technology, China, 1997), 58-78. (Chinese)

[22] T. J. Holmquist, G. R. Johnson and W. H. Cook, A Computational Constitutive Model for Concrete Subjected to Large Strains, High Strain Rates and High Pressures, 14th Int. Symp. Ballistics, Quebee, Canada, 1993, 591-600.

[23] S. T. Zalesak, Fully Multidimensional Flux-Corrected Transport Algorithms for Fluids, *J. Comput. Phys.* **31,** 335(1979).

[24] M. Sussman and E. G. Puckeett, A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows, *J. Comput. Phys.* **162,** 301(2000).

[25] J. H. Peng, A Study on High Velocity Impact, Master dissertation, (Beijing Institute of Technology, China, 1984) (Chinese).

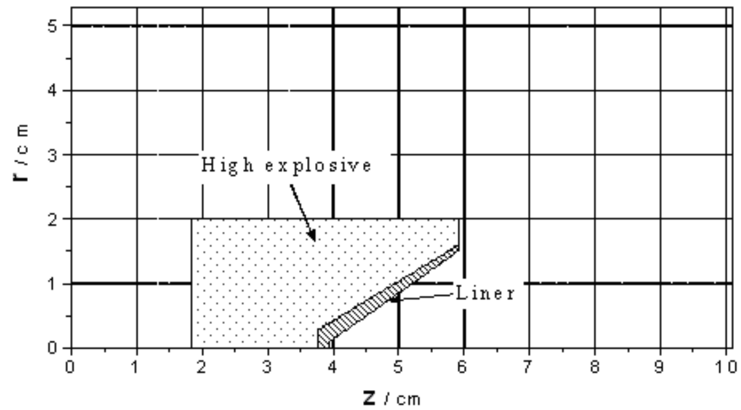
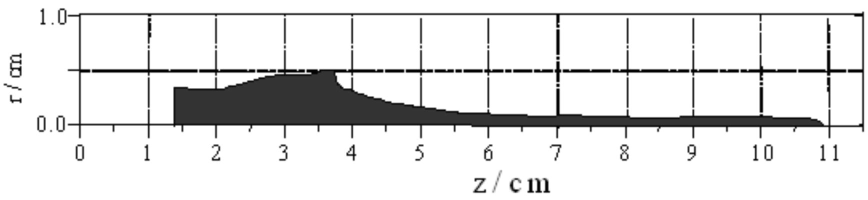
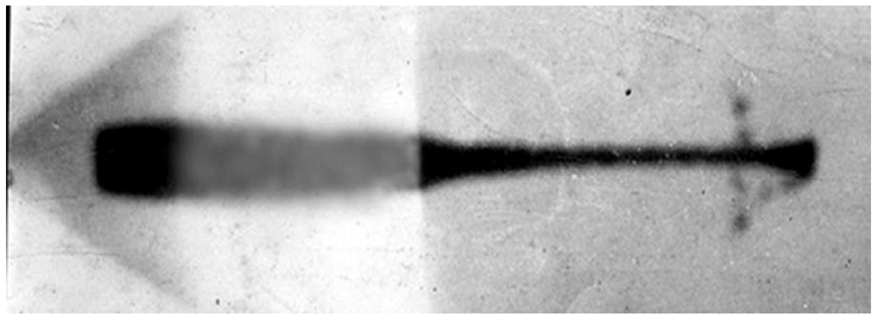


Fig. 5.6 Initial configuration of the shape charge



(a)



(b)

Fig. 5.7 Comparison with (a) the computational results at 30.004μs and
(b) the x-ray photograph (ratio: 1:1) at 30.0μs in the shaped charge problem

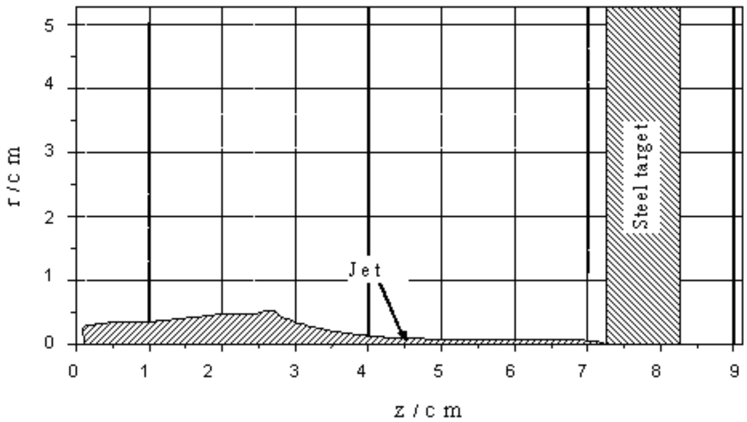
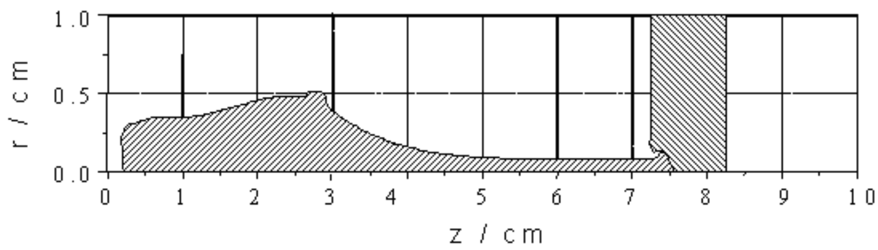
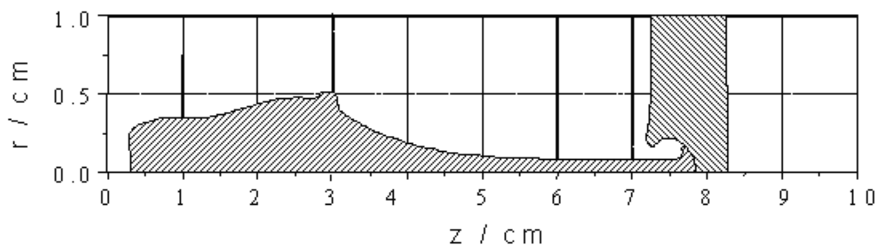


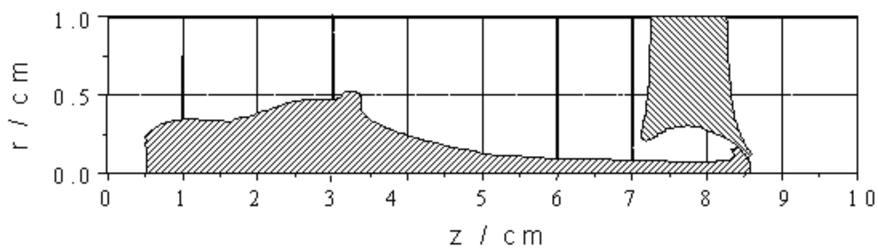
Fig. 5.8 Initial configuration of plate penetration



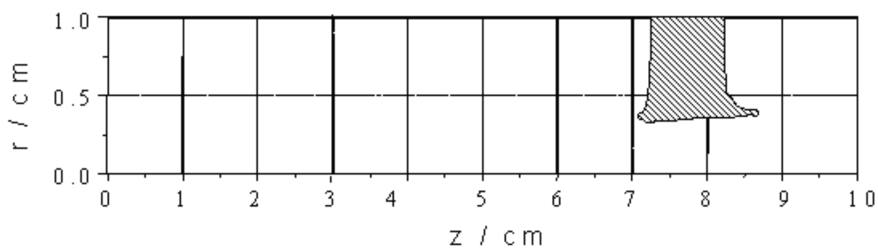
(a) Penetration at 1.571μs



(b) Penetration at 3.306μs



(c) Perforation at 6.809μs



(d) Forming steady hole at 11.917μs

Fig. 5.9 Penetration of a steel plate by a jet

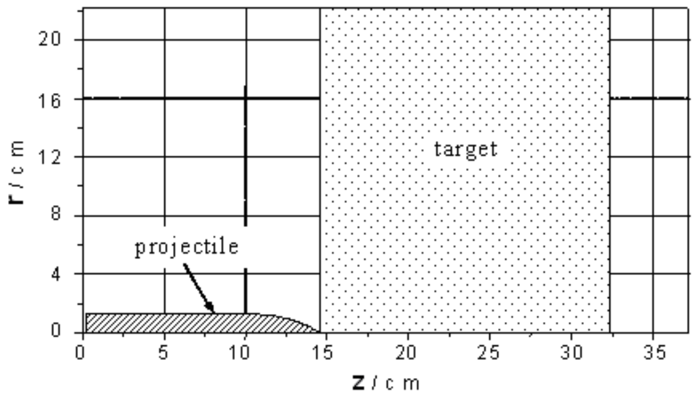
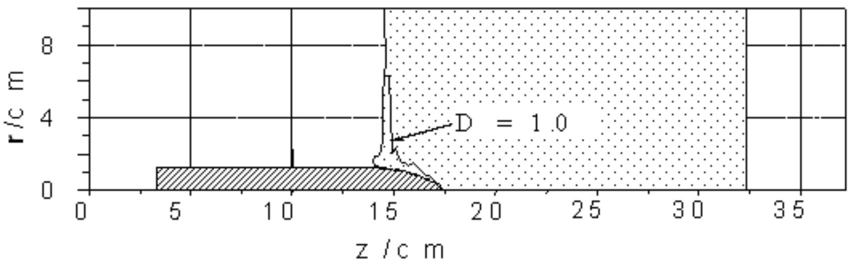
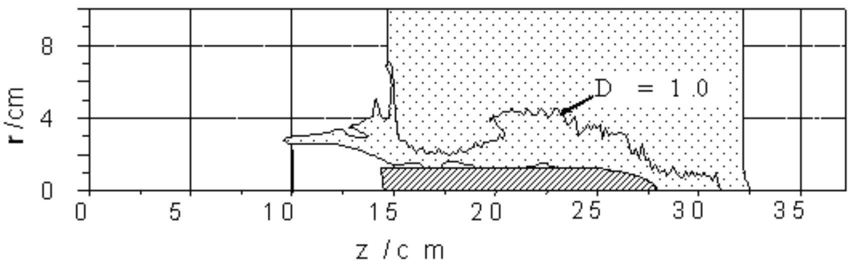


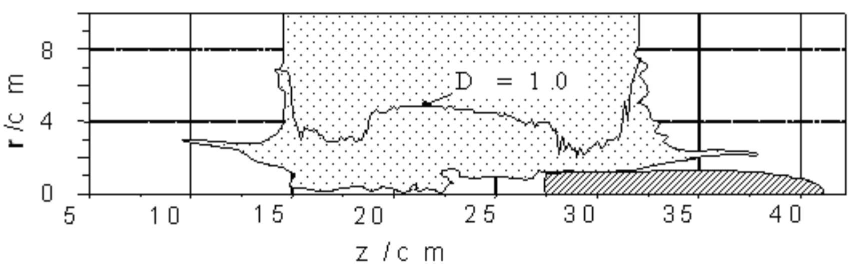
Fig. 5.10 Initial configuration for penetration with ogival nose projectile



(a) Penetration at 80.370μs



(b) Penetration at 424.395μs



(c) Penetration at 1023.371μs

Fig. 5.11 Penetration of concrete target by an ogival nose projectile

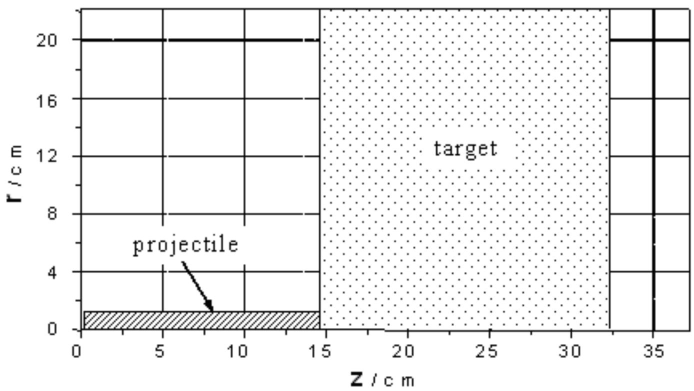
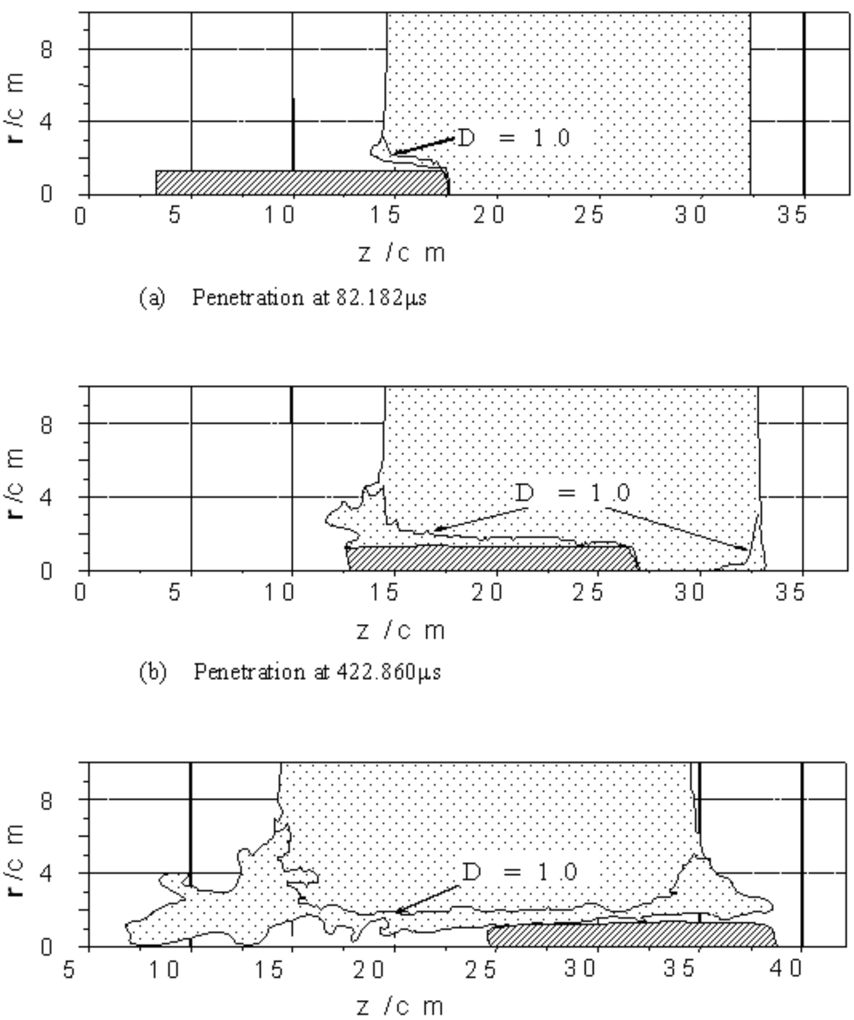


Fig. 5.12 Initial configuration for penetration with flat nose projectile



(a) Penetration at 82.182μs

(b) Penetration at 422.860μs

(c) Penetration at 1103.461μs

Fig. 5.13 Penetration of concrete target by a flat nose projectile