Two-Step Hybrid Remapping (Conservative Interpolation) for Multimaterial Arbitrary Lagrangian-Eulerian Methods LA-UR-10-05438

Markus Berndt[a], Jérôme Breil[b], Stéphane Galera[c], Milan Kucharik[d], Pierre-Henri Maire[e], Mikhail Shashkov*[f]

[a]*Los Alamos National Laboratory, CCS-2, Los Alamos, NM, 87545, U.S.A*
[b]*UMR CELIA, Université Bordeaux I 351, Cours de la Libération, 33 405 Talence, France*
[c] *INRIA, Team Bacchus, 351 Cours de la Libération, 33405 Talence Cedex, France*
[d]*Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Brehova 7, Praha 1, 115 19, Czech Republic*
[e]*CEA-CESTA BP 2, 33 114 Le Barp France*
[f]*Los Alamos National Laboratory, XCP-4, Los Alamos, NM, 87545, U.S.A*

Abstract

We present a new hybrid remapping (conservative interpolation) algorithm for multimaterial Arbitrary Lagrangian-Eulerian (ALE) methods. The hybrid remapping is performed in two steps. In the first step, only nodes of the grid that lie inside subdomains occupied by single materials are moved. At this stage, computationally cheap swept-region remapping is used. In the second step, nodes that are vertices of mixed cells (cells containing several materials) and vertices of some cells in the buffer zone around mixed cells are moved. At this stage, intersection-based remapping is used. This new approach results in an improved efficiency of remapping for multimaterial ALE methods. We demonstrate the performance of our new method for both structured and unstructured polygonal grids in two dimensions, as well as for cell-centered and staggered discretizations.

Key words: multimaterial Arbitrary Lagrangian-Eulerian methods, hybrid remapping, conservative interpolation
2000 MSC: 65M06, 65Z05

1. Introduction

In numerical simulations of fluid flow, the choice of the computational grid is crucial. Traditionally, there have been two viewpoints, utilizing the Lagrangian or the Eulerian framework, each with its own advantages and disadvantages. In a pioneering paper [10], Hirt et al. developed the formalism for a grid whose motion could be determined as an independent degree of freedom, and showed that this general framework could be used to combine the best properties of the Lagrangian and Eulerian methods. This class of methods has been termed Arbitrary Lagrangian-Eulerian or ALE.

*Corresponding author
Email addresses: berndt@lanl.gov (Markus Berndt), breil@celia.u-bordeaux1.fr (Jérôme Breil), Stephane.Galera@inria.fr (Stéphane Galera), kucharik@newton.fjfi.cvut.cz (Milan Kucharik), maire@celia.u-bordeaux1.fr (Pierre-Henri Maire), shashkov@lanl.gov (Mikhail Shashkov*)

Many authors have described ALE strategies to optimize accuracy, robustness, or computational efficiency, see for example [2, 3, 24, 13, 14, 27, 23].

For multimaterial flows it is common to separate the ALE scheme into three distinct stages. These are: 1) a Lagrangian stage in which the solution and the grid are updated; 2) a rezoning stage in which the nodes of the computational grid are moved to a more optimal position; and 3) a remapping stage, in which the Lagrangian solution is interpolated onto the rezoned grid.

For multimaterial flows, the initial grid is usually aligned with material interfaces, that is, each cell of the grid contains only one material. For simple flows, it is possible to rezone the grid in each material and keep material interfaces aligned with the grid at later times by not moving nodes on interfaces at all or moving them along interfaces during the rezoning stage of ALE. Due to the nature of shock wave propagation in complex high-speed multimaterial flows with strong shear deformations, ALE methods are currently the only proven technology for solving such problems.

In ALE methods, the grid does not move with the fluid, and for complex flows it is impossible to keep nodes of the grid on interfaces between materials during the entire calculation. Therefore, it is unavoidable that mixed cells containing two or more materials will appear. Mixed cells in ALE methods represent material interfaces that undergo high deformation.

The main problem related to mixed cells in the Lagrangian phase is how to accurately determine the thermodynamic states of the individual material components and the nodal forces that such a zone generates, despite the lack of information about the velocity distribution within multimaterial cells. Usually, a separate set of material properties is maintained for all the materials in each multimaterial cell along with the volume fractions that define the fraction of the cell's volume occupied by each material. A sub-scale model is then required to define how the volume fractions and states of the individual materials evolve during the Lagrangian step. The construction of such a model is beyond the scope of this paper. We refer the interested reader to [30] for more information and appropriate references. For the purpose of this paper, it is important to note that the accuracy of the closure model for a mixed cell depends on the accuracy of the information about each material, its parameters, as well as the material location inside the mixed cell.

In the rezoning stage, the nodes of the computational grid are moved to more optimal positions. The rezoning stage results in the new grid. We assume that the rezoned grid is close to the Lagrangian grid. In particular, we assume that, after rezoning, a node of the grid stays in the union of Lagrangian cells that share this node. The interested reader can find a short review of rezoning methods in [15].

To start a new Lagrangian step, we need to conservatively interpolate all flow parameters from the Lagrangian grid at the completed time step to the new rezoned grid. The assumption of small movements of nodes during the rezoning stage implies that an exchange of information during the remapping stage occurs only between immediate neighbors.

As we have mentioned before, realistic multimaterial ALE calculations have to deal with mixed cells, which may appear and disappear after the rezoning stage. Therefore, in the remap stage one needs to determine which cells of the rezoned grid are pure and which are mixed, and find parameters for each material in mixed cells. There are several approaches to multimaterial remap, [3], however, modern methods require knowledge of the interface on the Lagrangian grid.

It is beyond the scope of this paper to discuss interface reconstruction methods for the multimaterial case. We refer the interested reader to [1] for a review of such methods. For the purpose of this paper it is sufficient to know that if a mixed cell is a convex polygon in 2D (in this paper we are only concerned with 2D), then we assume each material in the mixed cell is represented by a convex sub-polygon, which contains only this material. If a mixed cell is non-convex, then a material in the mixed cell can be represented by disjoint pieces. However, each of these pieces is a convex polygon. These sub-polygons do not intersect with each other, and the union of all sub-polygons representing all materials is equal to the entire mixed cell.

To accurately represent materials on the new grid one needs to intersect cells of the new grid with pure sub-polygons representing materials on the old grid. This intersection can be computationally quite expensive, especially if non-convex cells need to be processed, which can result from the Lagrangian step. However, if a new cell is located inside a single pure material, then simple and cheaper methods, which do not require intersections, can be used for remapping this cell. One such relatively cheap method is based on the notion of a swept-region, [25].

The goal of this paper is to describe a new efficient hybrid remapping method to be used in multimaterial ALE simulations. This hybrid method uses a combination of computationally cheap swept-region remapping methods for cells inside sub-domains containing a single pure material, and intersection-based remapping for sub-domains containing mixed cells plus cells in some buffer region around those mixed cells. Hybrid remapping is performed in two steps. In the first step, only nodes of the grid inside sub-domains occupied by single materials are moved. In this stage, computationally cheap swept-region remapping is used. In the second step, nodes are moved that are vertices of mixed cells (cells containing several materials) and vertices, which are shared by pure cells containing different materials. At this stage, intersection-based remapping is used. In general our new approach improves the efficiency of remapping for multimaterial ALE codes while retaining the accuracy of intersection-based remapping. The general discussion related to factors affecting relative efficiency of remapping in multiphysics ALE codes is presented in section 7.

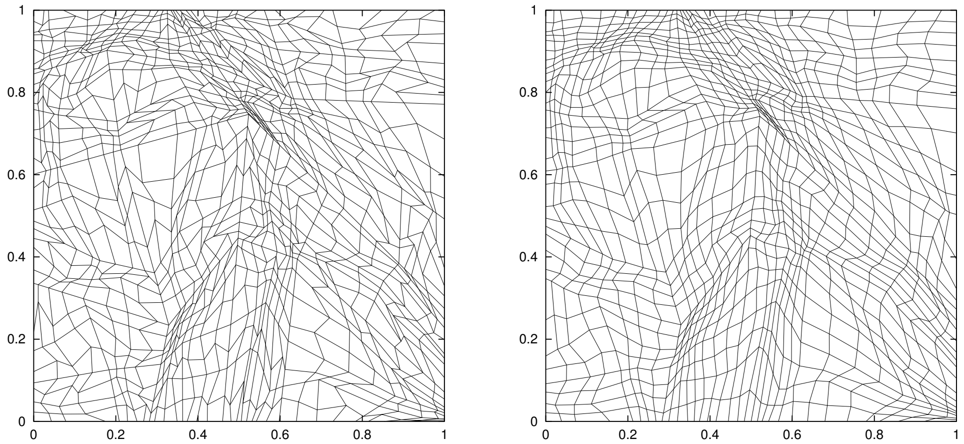
The paper is organized as follows. In section 2, we describe notations related to grids and representation of the materials. In section 3, we describe the main ideas of remapping for the single material case. We describe intersection-based remapping as well as swept-region-based remapping and describe their properties including a relative cost estimate. Conventional remapping for multimaterial ALE, which requires intersections, is described in section 4. The new hybrid remapping is described in section 5. In that section, we give a motivation, describe the algorithm and comment on the relative cost of hybrid remapping. Numerical experiments are presented in section 6. We present results using two different multimaterial ALE codes. The first code, [20, 17] uses structured quadrilateral grids and a staggered discretization. The second code, [9] uses general polygonal grids and a cell-centered discretization. We first present results for stand-alone cyclic remapping, where we know the exact solution. Then we present results for two realistic problems: a Rayleigh-Taylor instability problem and a problem of shock/bubble interaction. For all problems, we present timings using both ALE codes. We conclude with a discussion in section 7.

2. Notations

2.1. Grids

We consider a two-dimensional computational domain Ω, assumed to be a general polygon. We assume that we are given a grid on Ω that consists of cells C_i, $i = 1, \ldots, i_{max}$ that cover Ω without gaps or overlaps. The cells C_i can be non-convex.

Each cell is defined by a set of vertices (which we will sometimes call points or nodes), denoted by $P\left(C_i\right)$, and a set of sides (which are segments of straight lines, we will sometimes refer to them also as faces or edges), denoted by $F\left(C_i\right)$. Each side F_k is shared by only two cells, denoted by



a) b)

Figure 1: *a) Old (Lagrangian) grid, b) New (rezoned) grid.*

$C\left(F_k\right)$. Each vertex P_m may be shared by an arbitrary number of cells. We denote the set of cells that have a common vertex by $C\left(P_m\right)$; similarly, we denote the set of all sides sharing a common vertex P_m by $F\left(P_m\right)$. The cells that share a side or vertex with a particular cell are called *neighbors*; the set of all the neighbors of a cell C_i is denoted by $C\left(C_i\right)$. To distinguish between neighbors which share a face or a point, we will denote them by $C_f\left(C_i\right)$ and $C_p\left(C_i\right)$, respectively. The reciprocal relation of the neighborhood defines the *connectivity* of the grid.

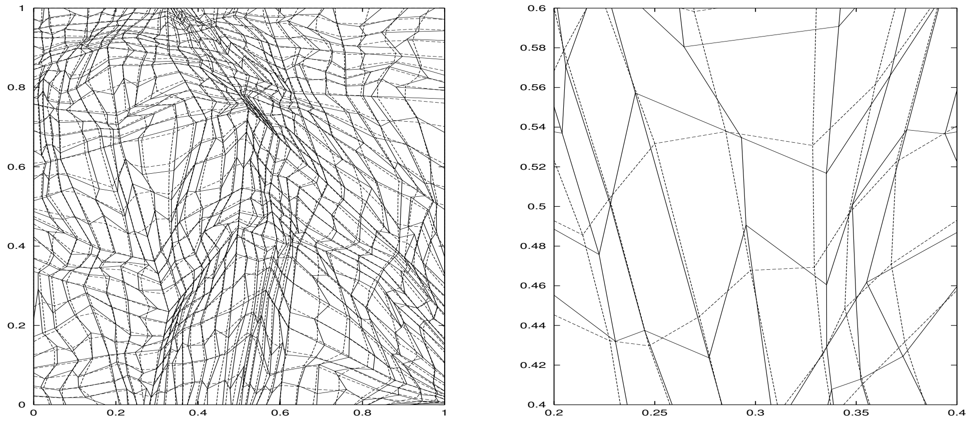
In the context of ALE methods, we consider two grids with the same connectivity — i.e., the same number of cells and vertices, and the same neighbor relations. The grid that contains the cells C_i is called the Lagrangian or *old* grid. The second grid, containing the cells $\tilde{C}_k$, is called the rezoned or *new* grid.

In the ALE method, the rezoned grid results from an algorithm (i.e., a rezoner) that identifies and mitigates inadequacies of the Lagrangian grid. In Figs. 1 and 2, we show examples of pairs of a Lagrangian grid and a rezoned grid. The rezoned grids were generated using the optimization-based reference Jacobian strategy described in [15]. The rezoned grid produced by this algorithm remains "close" to the Lagrangian grid, but has better geometrical quality. Fig. 2 illustrates how complicated the relative locations of the two grids can be, even when displacements of the nodes are small.

After rezoning, the old grid $\{C_i\}$ is mapped into a new grid $\{\tilde{C}_i\}$. We define a set $\mathcal{C}\left(C_i\right)=\bigcup_k C_k$, such that

$$\tilde{C}_i \in \mathcal{C}\left(C_i\right) . \tag{1}$$

For any two grids, such a set exists because $\tilde{C}_i \in \bigcup_{k=1}^{i_{max}} C_k$. However, we will always consider the



a) b)

Figure 2: *Lagrangian (solid lines) and rezoned (dashed lines) grid. a) Entire region, b) Fragment.*

reasonably small set for which (1) holds.

In this paper, we will assume that $\mathcal{C}\left(C_i\right) = C_i \bigcup C\left(C_i\right)$; that is, the new cell $\tilde{C}_i$ is contained in the union of the old cell C_i and its immediate neighbors, see, for example, Fig. 2b).

2.2. Representation of materials

In multimaterial problems each material is assigned some unique number k_m from the global list of materials $\mathcal{K} = \{k_m = 1,\, 2\,, \ldots\, k_{max}\}$.

Each cell in the grid can be pure, containing just one material, or mixed, containing several materials.

In this paper, we assume that materials are represented by pure sub-polygons containing only one material. This assumes that some interface reconstruction has been performed.

Fig. 3 a) is an example of a grid for a two-material problem. Pure cells containing only material 1 are in dark grey, and pure cells containing only material 2 are in white. In mixed cells containing both materials the boundary of the sub-polygons containing material 1 is marked by a red dashed line. Each material, k, in mixed cell C_i occupies part of the cell, which we denote by C_i^k. Each mixed cell C_i has a list $\mathcal{K}_i = \{k_1,\, k_2,\, \ldots\, ; k_s \in \mathcal{K}\}$ of materials, where k_i is one of the materials from the global list. In Fig. 3 b), we show one mixed cell which consists of four different materials. For a non-convex cell C_i, C_i^k can consist of several disjoint pieces, as material 2 in (Fig. 3 c)); and the material can be represented by a non-convex polygon, as material 1 in (Fig. 3 c)) (see [1] for details). It is important that the pieces representing all materials do indeed cover the mixed cell without gaps and overlaps. The presence of several disjoint pieces and non-convex polygons can affect the efficiency of the multimaterial remap.

If C_i^k consists of several disjoint pieces, then we will assume that we know the geometry of each such piece. For each material k we also know the total mass of the material, m_i^k. We will denote the volume of region C_i^k by $|C_i^k|$ or V_i^k, and the density of material k by $\bar{\rho}_i^k$.

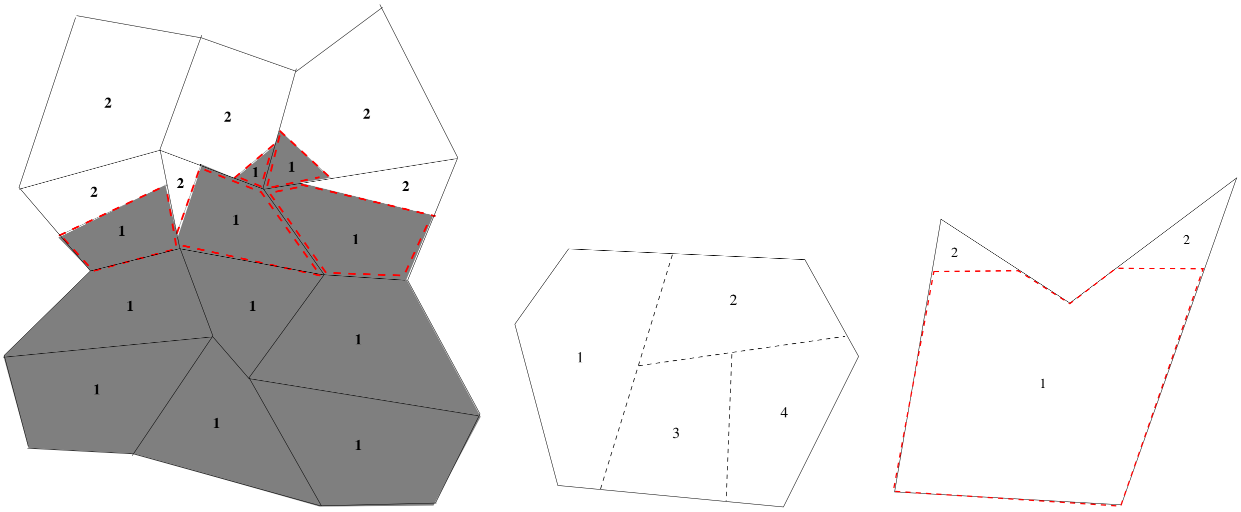


Figure 3: Representation of materials: a) Two-material case, in mixed cells boundaries of polygons representing material #1 are marked by a thick dashed red line. Material ids are inside pure cells and inside polygons representing materials in mixed cells; b) one mixed cell containing four different materials - Material ids are inside polygons representing materials; c) one non-convex mixed cell containing two materials - material #1 in represented by one non-convex polygon, and material #2 is represented by two triangles.

3. Remapping for the Single-Material Case

3.1. Statement of the Remapping Problem

We start with a definition of the remapping problem for a single material. Here we assume that there is a positive function $\rho(\vec{\mathbf{r}}) > 0$, $\mathbf{r} = (x, y)$, which we call density, that is defined throughout the problem domain. The only information that we are given about this function is its mean value in each of the cells of the old grid:

$$\bar{\rho}_i = \frac{\int_{C_i} \rho(\mathbf{r})\, dV}{V(C_i)}\,, \tag{2}$$

where $V(C_i)$ is the volume of the cell C_i. The numerator of (2) is the cell mass

$$m_i \equiv \int_{C_i} \rho(\mathbf{r})\, dV\,, \tag{3}$$

and so the mean density is

$$\bar{\rho}_i = \frac{m_i}{V(C_i)}\,. \tag{4}$$

The total problem mass is

$$M \equiv \int_{\Omega} \rho(\mathbf{r})\, dV = \sum_{i=1}^{i_{max}} \int_{C_i} \rho(\mathbf{r})\, dV = \sum_{i=1}^{i_{max}} m_i = \sum_{i=1}^{i_{max}} \bar{\rho}_i\, V(C_i)\,. \tag{5}$$

The problem statement is to find accurate approximations $\tilde{m}_i$ for the exact masses, m_i^{ex}, of the new cells

$$\tilde{m}_i \approx m_i^{ex} = \int_{\tilde{C}_i} \rho(\mathbf{r})\, dV\,. \tag{6}$$

The issue is to define what is meant by "accurate", since the underlying density field is not known in detail.

The approximate mean values of density in the new cells are defined by

$$\tilde{\bar{\rho}}_i = \frac{\tilde{m}_i}{V\left(\tilde{C}_i\right)} \,. \tag{7}$$

The common accuracy requirement for remapping is formulated as *linearity preservation.* That is, if the underlying function $\rho(\mathbf{r})$ is a global linear function, the remap must be exact;

$$\tilde{m}_i = m_i^{ex} \,. \tag{8}$$

Another important property of remapping is bound preservation. If we assume that a new cell is contained in the immediate neighborhood of the old cell, then it is natural to require

$$\min_{i' \in \mathcal{C}(C_i)} \bar{\rho}_{i'} \leq \tilde{\bar{\rho}}_i \leq \max_{i' \in \mathcal{C}(C_i)} \bar{\rho}_{i'} \,. \tag{9}$$

Finally, a statement of *global conservation* is formulated as

$$\sum_{i=1}^{i_{max}} \tilde{m}_i = M \,. \tag{10}$$

We will refer to the problem of finding accurate, bounded approximations for the masses and the corresponding mean densities on the new grid, such that total mass is conserved, as *bound-preserving conservative remapping (interpolation).*

3.2. Intersection-based Remapping

3.2.1. Aggregated Intersection-based Remapping

Each cell of the new grid $\tilde{C}_i$ is formed from pieces of the cells of the old grid C_i

$$\tilde{C}_i = \bigcup_{i'=1}^{i_{max}} \left(\tilde{C}_i \bigcap C_{i'}\right) = \bigcup_{i' \in \mathcal{C}(C_i)} \left(\tilde{C}_i \bigcap C_{i'}\right) \,. \tag{11}$$

The most natural approach to remapping can be based on this representation of the new cell:

$$m_i^{ex} = \int_{\tilde{C}_i} \rho(\mathbf{r})\, dV = \sum_{i' \in \mathcal{C}(C_i)} \int_{\tilde{C}_i \bigcap C_{i'}} \rho(\mathbf{r})\, dV \,. \tag{12}$$

The remapping based on this formula would be exact if we knew the density function everywhere on the old grid. However, as pointed out earlier, we only know the average value of $\rho(\mathbf{r})$ within a cell. Thus, it is necessary *to reconstruct* the density function in each cell of the old grid. Usually, this is a piece-wise linear reconstruction over cells of the old grid. This reconstruction involves an estimate of the gradient of the function on the cell and some limiting procedure for the gradient to avoid the creation of new extrema. Details of the reconstruction are not important for this paper and can be found elsewhere (see, for example, [21]). For the purpose of this paper, it is important to know that there is some cost associated with the reconstruction.

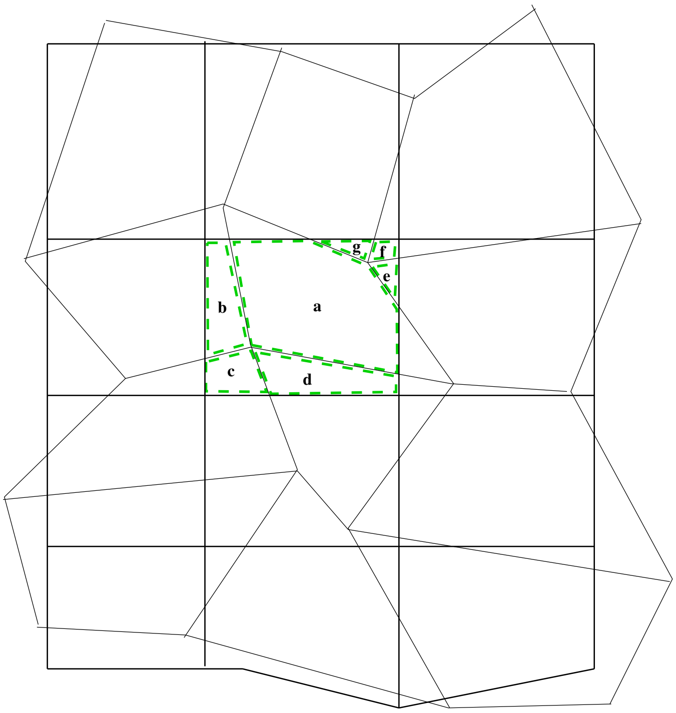


Figure 4: Aggregated-intersection-based remapping. The old grid is shown by thin solid lines, new grid is thick solid lines. Boundaries of intersection polygons, a)-g), corresponding to $\tilde{C}_i \bigcap C_{i'}$ are marked by dashed lines.

We denote the reconstruction over cell C_i as $\rho_i(\mathbf{r})$. For aggregated intersection-based remapping, this must be a *conservative* reconstruction

$$\int_{C_i} \rho_i(\mathbf{r})\, dV = \bar{\rho}_i\,. \tag{13}$$

Then, we can use the following approximation

$$\int_{\tilde{C}_i \bigcap C_{i'}} \rho(\mathbf{r})\, dV \approx \int_{\tilde{C}_i \bigcap C_{i'}} \rho_{i'}(\mathbf{r})\, dV\,. \tag{14}$$

This leads to the following formula for the remapped masses on the new grid

$$\tilde{m}_i = \sum_{i' \in \mathcal{C}(C_i)} \int_{\tilde{C}_i \bigcap C_{i'}} \rho_{i'}(\mathbf{r})\, dV\,. \tag{15}$$

It is important to note that any polynomial function can be integrated *exactly* over a polygonal cell.

We refer to the remapping method of (15) as aggregated intersection-based (AIB) method. Here, *aggregated* refers to the fact that the mass of a new cell is obtained by collecting pieces of masses from old cells.

The AIB method is conservative because

$$\begin{aligned}\sum_{i=1}^{i_{max}} \tilde{m}_i &= \sum_{i=1}^{i_{max}} \left(\sum_{i' \in \mathcal{C}(C_i)} \int_{\tilde{C}_i \bigcap C_{i'}} \rho_k(\mathbf{r})\, dV \right) = \sum_{i=1}^{i_{max}} \left(\sum_{i'=1}^{i_{max}} \int_{\tilde{C}_i \bigcap C_{i'}} \rho_{i'}(\mathbf{r})\, dV \right) \\ &= \sum_{i'=1}^{i_{max}} \left(\sum_{i=1}^{i_{max}} \int_{\tilde{C}_i \bigcap C_{i'}} \rho_{i'}(\mathbf{r})\, dV \right) = \sum_{i'=1}^{i_{max}} \left(\int_{C_{i'}} \rho_{i'}(\mathbf{r})\, dV \right) = \sum_{i'=1}^{i_{max}} m_{i'} = M\,.\end{aligned}$$

AIB remapping also is bound and linearity preserving if the reconstructed function $\rho_k(\mathbf{r})$ satisfies these properties, [25].

We note that AIB methods are very general and in principle can be applied when the old and new grids are not related to each other — they may even have a different number of cells of arbitrary shapes and a different connectivity.

We now summarize the main stages of AIB remapping:

- Conservatively reconstruct the function on the old grid. This includes estimating and limiting the gradient using information from neighboring cells.
- Find intersections of old and new cells. If the old and new grid have the same topology and the new grid is obtained from the old grid by small displacements (as it is the case in this paper), then one only needs to find intersections of new cells with the corresponding old cell and its immediate neighbors. Even in 2D, if cells of the old or new grids, or both, are non-convex, then it may be more efficient to subdivide cells into triangles and to intersect triangles representing old and new cells. We believe that, in 3D, subdividing cells into tetrahedra is the only way of doing intersections, because cells in 3D always have non-flat faces.
- Integrate the reconstructed functions over intersections. This involves the computation of integrals of polynomial functions over polygons, which can be reduced to boundary integrals and computed exactly.

3.2.2. Flux-Intersection-based Remapping

When the old and new grids have the same connectivity (as we assume in this paper), a new cell can be represented as follows

$$\tilde{C}_i = C_i \bigcup \left(\bigcup_{i' \in \mathcal{C}'(C_i)} \tilde{C}_i \bigcap C_{i'} \right) \setminus \left(\bigcup_{i' \in \mathcal{C}'(C_i)} C_i \bigcap \tilde{C}_{i'} \right), \tag{16}$$

where

$$\mathcal{C}'\left(C_i\right) = \mathcal{C}\left(C_i\right) \setminus C_i\,, \tag{17}$$

and where $\setminus$ is the difference operation on sets. In words, the new cell is the old cell plus pieces of neighboring cells that are added minus pieces of the old cell lost to other new cells. The corresponding representation for the exact mass of the new cell is

$$\begin{aligned}m_i^{ex} = \int_{\tilde{C}_i} \rho(\mathbf{r})\, dV &= \int_{C_i} \rho(\mathbf{r})\, dV + \sum_{C_{i'} \in \mathcal{C}'(C_i)} \int_{\tilde{C}_i \bigcap C_{i'}} \rho(\mathbf{r})\, dV - \sum_{C_{i'} \in \mathcal{C}'(C_i)} \int_{C_i \bigcap \tilde{C}_{i'}} \rho(\mathbf{r})\, dV \\ &= m_i + \sum_{C_{i'} \in \mathcal{C}'(C_i)} \mathcal{F}_{i,i'}^{ex}\,, \end{aligned} \tag{18}$$

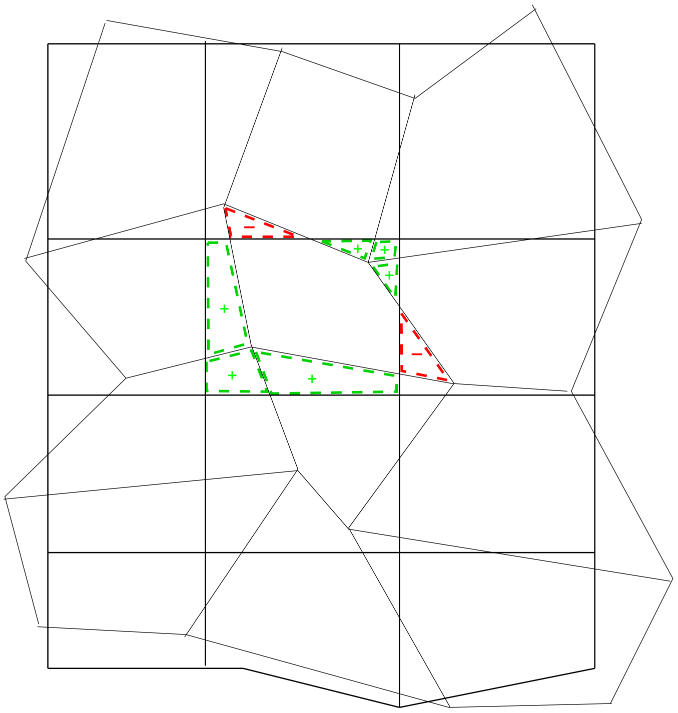


Figure 5: Flux-intersection-based remapping. The old grid is shown by thin solid lines and the new grid is shown by thick solid lines. Boundaries of intersection polygons, are marked by dashed lines. Signs + and − correspond to positive and negative pieces of corresponding generalized fluxes. For example, the generalized flux corresponding to the edge shared with right cell has two pieces; and the flux corresponding to the left edge has only one piece.

where

$$\mathcal{F}^{ex}_{i,i'} = \int_{\tilde{C}_i \bigcap C_{i'}} \rho(\mathbf{r})\, dV - \int_{C_i \bigcap \tilde{C}_{i'}} \rho(\mathbf{r})\, dV \,, \tag{19}$$

are *generalized mass fluxes*. We note that the second cell indicated by the index i' may be any cell in the neighborhood and is not restricted to those with whom the cell C_i has a common side. Equation (18) is exact, and illustrates that the mass of the new cell can be written as the mass of the corresponding old cell plus the exchange of masses with neighboring cells.

The remapping based on representation (19) has the important theoretical advantage that it is conservative without requiring any properties in the reconstruction nor an exact integration of the reconstructed function. In fact, any formula of the form

$$\tilde{m}_i = m_i + \sum_{C_{i'} \in \mathcal{C}'(C_i)} \mathcal{F}_{i,i'} \,, \tag{20}$$

where $\mathcal{F}_{i,i'} = -\mathcal{F}_{i',i}$ is some approximation of the flux, will be conservative because of detailed balance.

We will refer to methods based on equation (20) and some approximation of (19) as flux-intersection-based (FIB) methods.

In Fig. 5, we schematically present positive and negative pieces of the generalized fluxes. We note that, if AIB and FIB methods are using the same reconstruction of ρ on the old grid and generalized fluxes in the FIB method are computed as

$$\mathcal{F}_{i,i'} = \int_{\tilde{C}_i \bigcap C_{i'}} \rho_{i'}(\mathbf{r})\, dV - \int_{C_i \bigcap \tilde{C}_{i'}} \rho_i(\mathbf{r})\, dV\,, \tag{21}$$

then the two methods are algebraically equivalent [1]. However, as far as their implementation is concerned, they can be quite different.

We now summarize the main stages of FIB remapping:

- As we have mentioned before in general, FIB remapping does not require conservative reconstruction. However, some reconstruction is still needed to estimate fluxes. To make the comparison with AIB remapping easier, we still assume that the reconstruction is conservative and it also includes estimating and limiting the gradient using information from neighboring cells. Hence, this step is exactly the same as for AIB remapping.

- Find intersections of old and new cells. In general, one needs to find two intersections per edge and four intersections per vertex of the cell.

- Integrate the reconstructed functions over intersections. This involves the computation of integrals of polynomial functions over polygons, which can be reduced to boundary integrals.

3.3. Swept-Region-based Methods

Representation (20) allows us to make simplifying approximations that will allow us to avoid the detailed calculation of the intersections between the cells of the new and old grids without sacrificing exact conservation.

The approximations that we use are based on two ideas, [25]. First, up to fourth-order accuracy, the exact masses of new cells can be represented as line integrals of polynomial functions over the boundary of a new cell. Second, the line integral over the boundary of the new cell is the line integral over the boundary of the old cell (which is the old mass) plus the line integrals over the regions swept by the movement of the faces (i.e., sides) of the cell C_i, Fig. 6. The face which shares cells C_i and $C_{i'}$ can be denoted by two indices, i, i'. The corresponding swept region is denoted by $\delta F_{i,i'}$.

We define swept-region-based remapping (SRB) as follows

$$\tilde{m}_i = m_i + \sum_{C_{i'} \in C_f(C_i)} \mathcal{F}_{i,i'}\,, \tag{22}$$

where

$$\mathcal{F}_{i,i'} = -\mathcal{F}_{i',i}\,.$$

In (22), fluxes correspond only to neighbors which share faces. The approximate ”fluxes” $\mathcal{F}_{i,i'}$ are

$$\mathcal{F}_{i,i'} = \int_{\delta F_{i,i'}} \rho_{i,i'}(\mathbf{r})\,, \tag{23}$$

[1] In the rest of this paper we always will be using definition (21) for generalized fluxes in FIB methods.

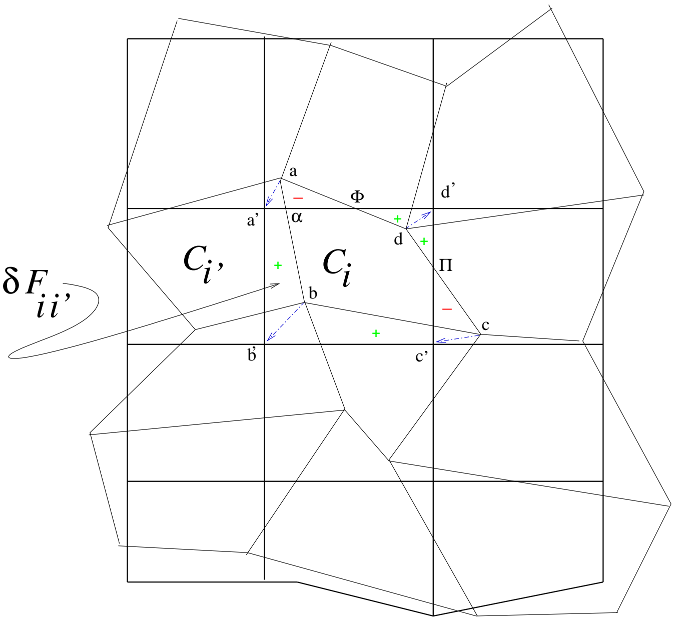


Figure 6: Swept-region-based remapping for single material. Old cell $C_i = \{a, b, c, d\}$, and corresponding new cell $\{a', b', c', d'\}$. Swept region $\delta F_{i,i'}$ corresponding to face a, b (face shared by cells C_i and $C_{i'}$ - “left” cell) is quadrangle $\{a, a', b', b\}$, and it has positive signed volume. Swept region corresponding to face c, d is self-overlapping but still has positive signed volume because the “positive” triangle $\{c', c, \Pi\}$ is bigger than the “negative” triangle $\{\Pi, d, d'\}$. Triangle $\{a', \alpha, a\}$ belongs to two swept regions corresponding to faces a, b and d, a, but its contribution to the area of these swept regions has a different sign.

where the density function associated with the faces is defined depending on the sign of the volume of the swept region[2]

$$\rho_{i,i'} = \begin{cases} \rho_{i'}(\mathbf{r}) \,, & V(\delta F_{i,i'}) \geq 0 \,, \\ \rho_i(\mathbf{r}) \,, & V(\delta F_{i,i'}) < 0 \,, \end{cases} \,, \tag{24}$$

where $V(\delta F_{i,i'})$ is the signed volume of the swept region.

The main advantage of the SRB method (23), (24) is that it does not require finding intersections of the old and new grids, which makes it computationally much cheaper when compared to intersection-based methods.

We now summarize the main stages of SRB remapping:

- Conservatively reconstruct the function on the old grid, which includes estimating and limiting gradient using information from neighboring cells.[3]
- Integrate the reconstructed functions over swept regions. The computation of integrals of polynomial functions over polygons can be reduced to boundary integrals.

[2] The signed volume is defined using a line integral representation of the volume where orientation of the boundary of the swept region is taken in such way that the signed volume is positive if the swept region is added to the cell and negative other-wise, [25].

[3] As for any flux based methods this is not required for conservation of the overall method (see note related to FIB method).

The detailed derivation and analysis of swept-region-based methods is presented in [25], [21].

3.4. Properties of Remapping Methods for a Single Material

Both intersection-based and swept-region based methods are linearity-preserving, if the reconstruction procedure for $\rho_i(\mathbf{r})$ is linearity-preserving. The formal order of accuracy (the accuracy of remapping for smooth functions) is the same for both methods. In particular, both methods exactly compute volumes of cells of the new grid, which corresponds to setting $\rho_i(\mathbf{r}) = 1$. Hence, they satisfy the geometric conservation law (GCL), [8].

Clearly, intersection-based methods are the most accurate methods for a given reconstruction of the function on the old grid. The remapped value in the new cell depends on all values in the neighboring cells. In particular, it involves values in neighboring cells which are sharing only a vertex with the cell under consideration. This can be significant, if the underlying function is discontinuous and values of the function in neighbor cells that are only vertex connected are orders of magnitude different from the value in the cell under consideration. Swept-region-based methods are using information only from face connected neighbor cells and can be less accurate in such situations.

However, it is interesting to note that results presented in [21], show that for single material cyclic remapping examples (that is, repeated remapping of given function on a sequence of grids) the accuracy of swept-region-based and intersection-based methods is comparable even for discontinuous functions.

We now consider the relative computational cost of the methods for the example of logically rectangular grid with $N \times M$ cells. The total number of cells is then $N * M$, the total number of vertices is $(N+1) * (M+1)$ and the total number of edges is $N * (M+1) + M * (N+1)$. Let us also denote the cost of function reconstruction by C_{rec}, the cost of intersection by C_{inter} and the cost of integration by C_{integ}.

The total cost of the AIB method is

$$C_{AIB} = N * M * C_{rec} + N * M * \left[9 * (C_{int} + C_{integ})\right], \tag{25}$$

here we took into account that for a logically rectangular grid we need to intersect new cell $\tilde{C}_i$ with old cell C_i and its eight neighbor cells.

The total cost of the FIB method is

$$C_{FIB} = N * M * C_{rec} + 2 * \left[2 * (N+1) * (M+1) + N * (M+1) + M * (N+1)\right]\left[C_{int} + C_{integ}\right], \tag{26}$$

where the second term on the RHS (Right-Hand-Side) includes two intersections and the corresponding integration per face and vertex.

For fine grids, where we keep only terms proportional to $M * N$, we have

$$C_{AIB} \sim N * M * \left[C_{rec} + 9 * (C_{int} + C_{integ})\right] \tag{27}$$

and

$$C_{FIB} \sim N * M * \left[C_{rec} + 8 * (C_{int} + C_{integ})\right]. \tag{28}$$

Clearly, FIB remapping is cheaper than AIB remapping if we formally count number of arithmetic operations.

We now consider the SRB method. Its total cost is

$$C_{SRB} = N * M * C_{rec} + \left[(N+1) * M + (M+1) * N\right] * C_{integ}, \tag{29}$$

which for a fine grid means that

$$C_{SRB} \sim N * M * (C_{rec} + 2 * C_{integ}) \,. \tag{30}$$

In comparison with intersection based methods, the SRB method includes the same cost of function reconstruction on the old grid, but it does not include the intersection cost. Additionally, the number of integrations is reduced.

It is important to understand that, if a hydro code requires remapping of many physical quantities and not just density, then reconstruction must be done for all quantities but intersections can be done once and used for all quantities. Similarly, the most costly part of integration, which is the integration of 1, x, $y, xy, \ldots$ can be done once for all quantities. This observation suggests that the relative efficiency of remapping methods has to be analyzed for each specific situation.

An alternative remapping approach is presented in [11, 12], It can be considered an intermediate between intersection-based and swept-region-based methods. In this approach, if the swept region is self-overlapping (as, for example, the swept region corresponding to edge c, d in Fig. 6), then it is represented as two triangles. To find these two triangles one needs to intersect the old and new edge (which is much cheaper than the intersection of polygons). Each of these two triangles now is treated as a swept region by itself and the reconstructed function in the triangle is taken according to the sign of its signed volume.

4. Remapping for the Multimaterial Case

In this section, we describe conventional remapping methods for multimaterial ALE methods.

4.1. Statement of Remapping

In multimaterial remap, pure and mixed cells in the new grid must be identified. For each pure cell, the mass of the corresponding material must be computed. This is similar to the case of single material remap. For mixed cells, the mass of each material must be determined, as well as its volume, because after remap we may need to reconstruct interfaces on the new grid. We also may need some additional information. For example, the advanced interface reconstruction moment-of-fluid (MOF) method requires remapping of material centroids, [1, 7].

In the case of several materials the total volume $\mathcal{V}_k$ and total mass M_k of each material k must be conserved

$$\mathcal{V}_k = \sum_i V_i^k = \sum_i \tilde{V}_i^k = \tilde{\mathcal{V}}_k \,, \quad M_k = \sum_i m_i^k = \sum_i \tilde{m}_i^k = \tilde{M}_k \,, \tag{31}$$

where the sum is formally over all cells, but in practice can be taken only over pure cells containing material k and mixed cells containing this material.

Accuracy and monotonicity issues of multimaterial remap are beyond of the scope of this paper and will be addressed in the future.

In general, for multimaterial ALE one needs to use intersection-based remapping, as it is done, for example, in [35, 20, 17, 9].

4.2. Multimaterial Aggregated Intersection-based Remapping

In the multimaterial case, pieces of material k are collected in the new cell $\tilde{C}_i$ according to the following formula

$$\tilde{C}_i^k = \bigcup_{i' \in \mathcal{C}(C_i)} \left(\tilde{C}_i \bigcap C_{i'}^k \right) \,. \tag{32}$$

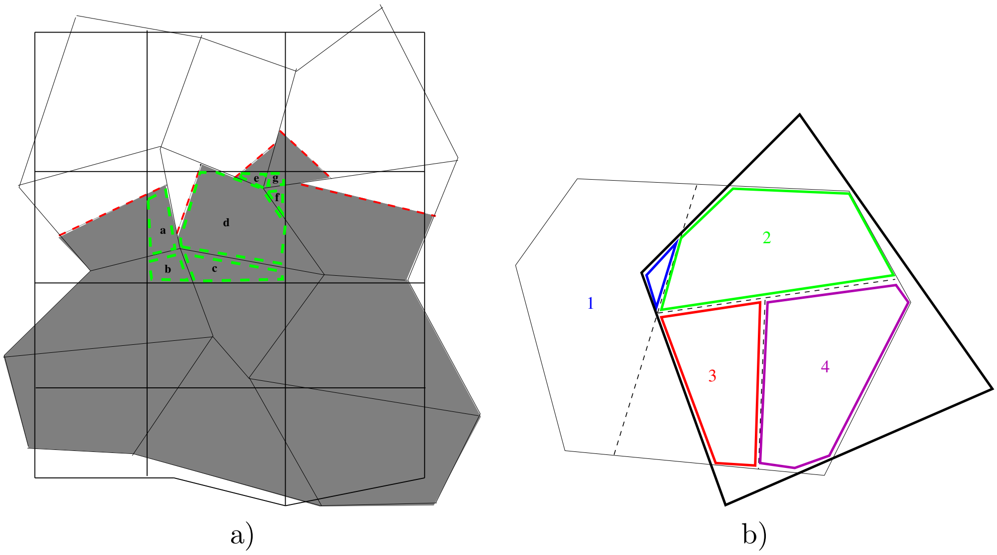


Figure 7: MAIB - Multimaterial remap AIB method. Old grid - thin solid lines, and new grid - thick solid lines: a) Two-material case - old and new grid. Pieces of material #1 from numerous old cells which contribute to central new cell are marked by a, b, c, d, e, f, g in mixed cells in mixed cells. Boundaries of polygons representing material #1 are marked by thick dashed red line. Material ids are inside pure cells and polygons representing material in mixed cells; b) one mixed cell containing four different materials - (material ids are inside polygons representing material) intersecting with new cell.

Equation (32) is similar to equation (11), except that new cell $\tilde{C}_i$ intersects with pure sub-polygons C_i^k. This process is illustrated in Fig. 7. We denote the method based on formula (32) by MAIB - multimaterial aggregated-intersection based.

The total volume of material k in new cell C_i^k is computed as the sum of volumes of corresponding intersections

$$|\tilde{C}_i^k| = \tilde{V}_i^k = \sum_{C_{i'} \in \mathcal{C}(C_i)} \int_{\tilde{C}_i \bigcap C_{i'}^k} dV\,. \tag{33}$$

The total mass of material k in new cell C_i^k is computed similarly

$$\tilde{m}_i^k = \sum_{C_{i'} \in \mathcal{C}(C_i)} \int_{\tilde{C}_i \bigcap C_{i'}^k} \rho_{i'}^k(\mathbf{r})\, dV\,, \tag{34}$$

where $\rho_{i'}^k(\mathbf{r})$ is some reconstruction of the density of material k in $C_{i'}^k \in C_{i'}$.

For advanced interface reconstruction, we may also need to know the centroids of each material in a mixed cell. The centroid is defined as the ratio of first and zeroth moment of the domain occupied by the material. The zeroth moment is simply the volume of the domain occupied by the material and it is defined by equation (33). The first moment, $\tilde{\mu}_i^k$, is defined as

$$\tilde{\mu}_i^k = \int_{\tilde{C}_i^k} \mathbf{r}\, dV = \sum_{C_{i'} \in \mathcal{C}(C_i)} \int_{\tilde{C}_i \bigcap C_{i'}^k} \mathbf{r}\, dV\,. \tag{35}$$

In the process of remapping, we have to identify if a new cell is pure or mixed. This requires additional logical operations, and building a list of materials for each new mixed cell, as well as identifying the situation when a mixed cell becomes pure. As a result of intersections, for each new mixed cell, we know volume, mass, and centroid of each material. In fact, we know exactly all pieces of pure polygons representing materials on the old grid, which are now in the new cell. In Fig. 7 a) these pieces are marked by a, b, c, d, e, f, g. The goal of interface reconstruction is to simplify this representation by replacing a collection of pieces by one or possibly several (in case of non-convex cell $\tilde{C}_i$) polygons, such that the volume of this polygon exactly equals the sum of volumes of the pieces.

We note that, again, as in the case of a single material all integrals in the equations in this section can be computed exactly, because they are integrals of polynomial functions over polygonal domains.

It is clear that, in comparison with the single material case, MAIB remapping requires more intersections — a new cell has to be intersected with all material polygons in its neighboring old cells.

However, the main stages of MAIB are similar to AIB. The total cost of MAIB can be roughly estimated by a formula similar to (25), where the total number of cells (which was $N * M$ for single AIB) is replaced by the total number of pure sub-polygons, and the number of neighbors is increased for mixed cells and for their neighbors. In conventional multimaterial ALE codes, MAIB remapping is used for the entire grid. Clearly, the cost of intersection based MAIB remapping is dominated by the cost of intersections, and it is much higher than the cost of SRB remapping for the single material case for the same grid.

4.3. Multimaterial Flux-Intersection-based Remapping

Formulas for multimaterial flux-intersection-based (MFIB) remapping are

$$\tilde{C}_i^k = C_i^k \bigcup \left(\bigcup_{C_{i'} \in \mathcal{C}'(C_i)} \tilde{C}_i \bigcap C_{i'}^k \right) \setminus \left(\bigcup_{C_{i'} \in \mathcal{C}'(C_i)} C_i^k \bigcap \tilde{C}_{i'} \right) , \tag{36}$$

$$\tilde{m}_i^k = m_i^k + \sum_{C_{i'} \in \mathcal{C}'(C_i)} \left(\int_{\tilde{C}_i \bigcap C_{i'}^k} \rho_{i'}^k(\mathbf{r}) \, dV - \int_{C_i^k \bigcap \tilde{C}_{i'}} \rho_i^k(\mathbf{r}) \, dV \right) , \tag{37}$$

$$\tilde{V}_i^k = V_i^k + \sum_{C_{i'} \in \mathcal{C}'(C_i)} \left(\int_{\tilde{C}_i \bigcap C_{i'}^k} dV - \int_{C_i^k \bigcap \tilde{C}_{i'}} dV \right) , \tag{38}$$

$$\tilde{\mu}_i^k = \mu_i^k + \sum_{C_{i'} \in \mathcal{C}'(C_i)} \left(\int_{\tilde{C}_i \bigcap C_{i'}^k} \mathbf{r} \, dV - \int_{C_i^k \bigcap \tilde{C}_{i'}} \mathbf{r} \, dV \right) . \tag{39}$$

When comparing the MAIB and MFIB methods we can make the same comments as at the end of section 4.2. That is, estimate (26) can be used for the cost of MFIB if one considers the more complicated unstructured grid where cells are pure sub-polygons and counts numbers of cells, edges and points in such a grid.

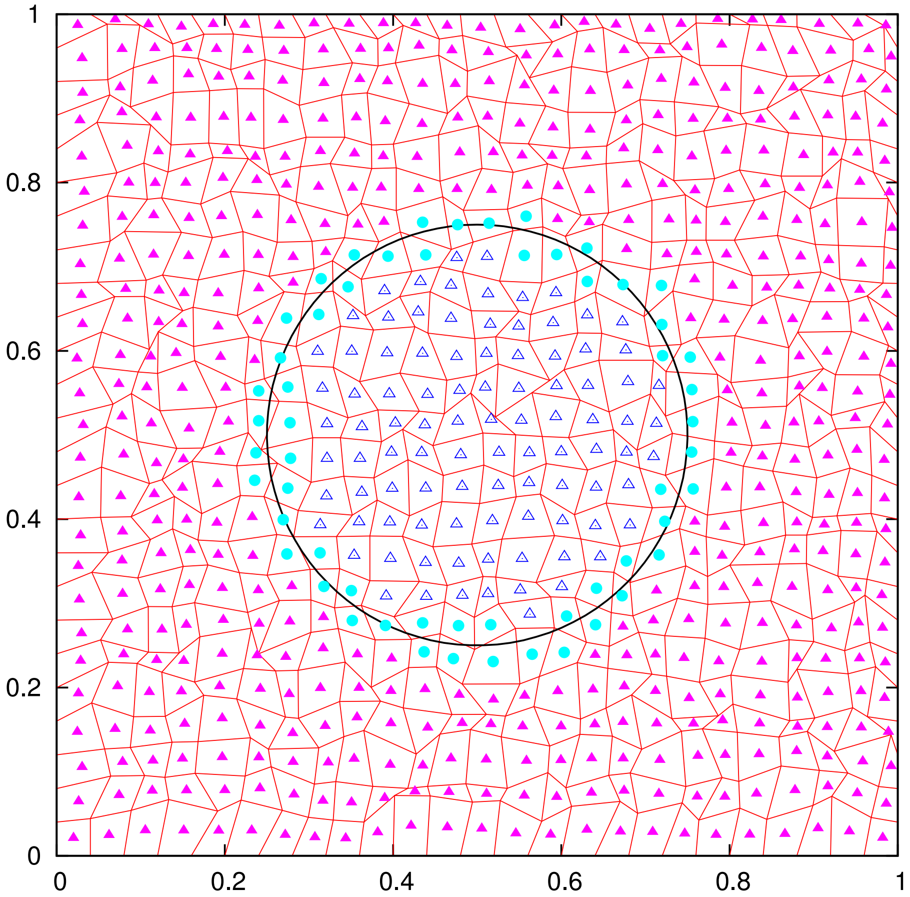


Figure 8: Two-material case. Lagrangian (old grid) - red solid lines; interface (circle) - solid black line; pure cells containing only material 1 are marked by blue triangles placed inside the cells; pure cells containing only material 2 are marked by red solid triangles placed inside the cells; mixed cells containing both material are marked by cyan solid circles.

5. Hybrid Multimaterial Remapping

5.1. Motivation

As motivation we consider the example with two materials depicted in Fig. 8. The computational domain is the unit square. Materials are separated by the circular interface with radius 0.25 and center (0.5, 0.5). Material 1 is inside the circle and material 2 is outside this circle. The computational grid is a distorted logically rectangular grid of $625 = 25 \times 25$ cells. The number of pure cells containing material 1 is 95 or 15.2% of the total number of cells. The number of pure cells containing material 2 is 474 or 75.84% of the total number of cells. The total number of pure cells is 569 or 91.04%, and the total number of mixed cells is 56 or 8.96%. Even for this coarse grid, the number of mixed cells is significantly smaller than the number of pure cells. In general, for the non-degenerate case, if the total number of cells is $\mathcal{N}$ then the number of mixed cells is proportional to $\sqrt{\mathcal{N}}$ and therefore the percentage of mixed cells approaches zero as the grid is refined. For example, if we refine the grid in the previous example and make the total number of cells equal to $\mathcal{N} = 160000 = 400 \times 400$ then the percentage of mixed cells will drop to only 0.5%. For this particular example, the number of mixed cells is approximately equal to $C_{mixed} * \sqrt{\mathcal{N}}$, where $C_{mixed} \sim 2$.

A straightforward approach for remapping in this multimaterial case is to use the MAIB or MFIB intersection-based methods described in the previous sections.

We now recall that intersection-based methods are quite expensive in comparison with swept-region-based methods because they require the intersection of the old and new grid. This fact and the observation that the percentage of mixed cells is small motivate us to construct a *hybrid* method, where ”inside” pure materials we use swept-region-based remapping and for mixed cells we use an intersection-based method. The cost of such a hybrid method can be expected to be between the cost of a swept-region-based method (which it approaches, when the grid is refined and the percentage of mixed cells goes to zero) and the cost of intersection-based remapping.

Recall that we assume vertices of the new grid to be located inside the union of the old cells sharing a corresponding vertex. It is due to this assumption that we can be sure that a pure cell is surrounded by pure cells of the same material. As a consequence, this cell will remain pure in the new grid and therefore we can use swept-region remapping to update the density in this cell.

For cells which are mixed or may become mixed we need to use intersection-based methods.

The main problem in constructing hybrid method is to decide what to do with fluxes corresponding to the edges, which are shared by pure and mixed cells.

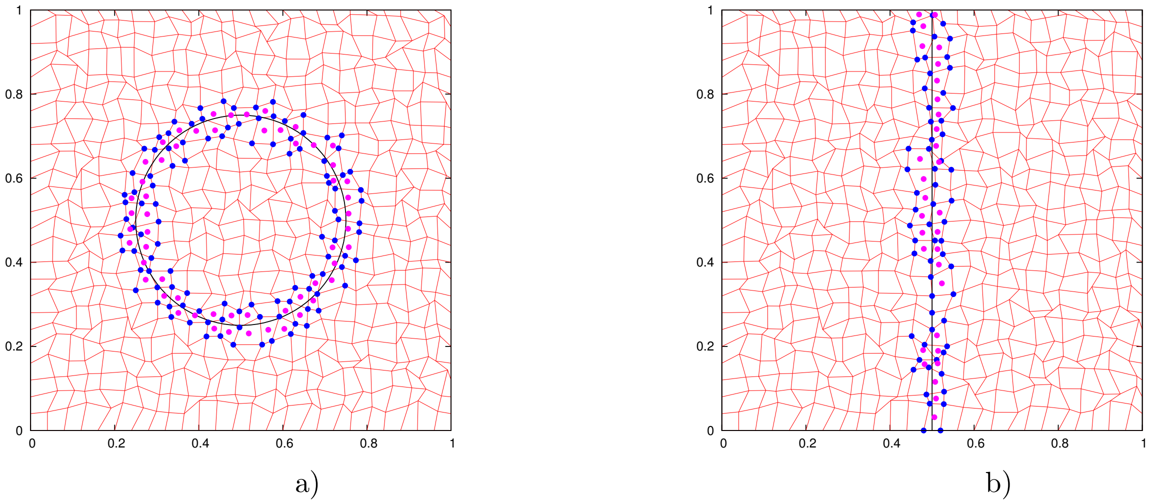
Intersection-based and swept-region based remapping methods cannot easily be combined, even in the single material case[4]. To explain the problem, we consider a hypothetical hybrid method which combines SRB, the flux-swept-region-based method, and FIB, the flux-intersection-based method. The first idea that comes to mind is to use the flux from the SRB or FIB method depending on the type of edge: Use the flux from the SRB method for edges that are shared by pure cells and use the the flux from the FIB method for edges that are shared by pure and mixed cells. Unfortunately, such an approach does not work for several reasons. First, the number of fluxes for the SRB and FIB methods are different; SRB has only edge related fluxes and FIB additionally has vertex based fluxes. Second, even edge based fluxes are of a different nature in either method. In the SRB method, an edge flux controls the exchange not only between cells sharing this edge, but also between cells sharing end points of this edge and the cell under consideration. In contrast, for the FIB method, the edge flux controls the exchange only with the cell sharing this edge. Thus, SRB and FIB edge fluxes refer to different geometric objects. One can be easily convinced that such a simple approach will not work by observing that replacing the SRB flux with the FIB flux for one of the edges will not be exact even for remapping of the constant density, $\rho = 1$, hence, it does not satisfy the geometric conservation law.

Now we will describe one possible approach to constructing a hybrid remapping method.

5.2. Pure and Mixed Points.

We start with a classification of points (vertices) of an old grid. We will call a point *mixed* in two cases: if the point is the vertex of a mixed cell or if the point is the vertex that is shared by pure cells and not all of these cells contain the same material; the rest of points are pure. Fig. 9 illustrates this definition.

[4] For some special cases like structured quadrilateral grids this can be done, [16].



a)

b)

Figure 9: Two-material case. Lagrangian (old grid) - red solid lines. Left panel - a): interface (circle) - solid black line; mixed cells containing both material are marked by magenta solid circles. Right panel - b) interface (straight line) - solid black line; mixed cells containing both material are marked by magenta solid circles; mixed points are marked by dark blue circles. In right panel there are several mixed points (in region close to (0.5, 0.3)), that lie on the interface and are shared by pure cells of different materials.

The main reason for such a classification of points is that the movement of *pure* points *cannot* create new mixed cells, Fig. 10. Therefore, if only pure points have been moved, remapping between the old grid and an *intermediate* grid obtained from the old grid by movement of only pure points can be performed by the SRB method.

In contrast, it is obvious that movement of *mixed* points *can* create new mixed cells.

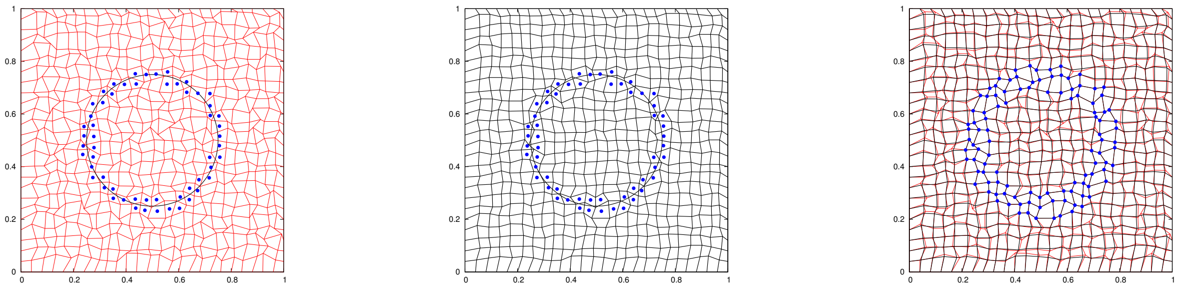
This observation suggests the following two-step rezoning/remapping algorithm.

5.3. Algorithm

5.3.1. Swept-Region Rezoning/Remapping Step.

In the first step, only pure points are moved (rezoned). For example, this can be done by using any rezoning algorithm, (cf., [32], [15]), while keeping mixed points fixed - this creates intermediate grid. We assume that the rezoning algorithm creates a valid grid. Examples of an old and an intermediate grid are presented in Fig. 10 a) and b), respectively. In this case, the intermediate grid is obtained by one Laplace smoothing step, which is just the simple average of coordinates of vertices of surrounding cells.

Remapping from the old to the intermediate grid is performed using the SRB method. In fact, this method can be applied to the entire grid, because edges of mixed cells and edges between pure cells containing different materials do not move (hence, corresponding fluxes are zero). Alternatively, we can mark such edges and not compute the corresponding fluxes which we know to be zero. As a result of the first step we have updated values of mass and density in all pure cells. Densities in mixed cells did not change. At this stage, interface reconstruction in mixed cells is not needed because mixed cells are not directly affected. Therefore, interfaces in mixed cells are the



a) b) c)

Figure 10: Left panel: old grid - solid red lines, interface and mixed cells. Central panel: only pure points have been moved, grid after movement of pure cells - solid black lines; no new mixed cells have been created. Right panel: overlap of old and new grid obtained by movement of only pure points; mixed points are marked by solid blue circles; mixed cells are the same for old and new grid.

same as they were on the old grid[5].

5.3.2. Intersection-based Rezoning/Remapping Step.

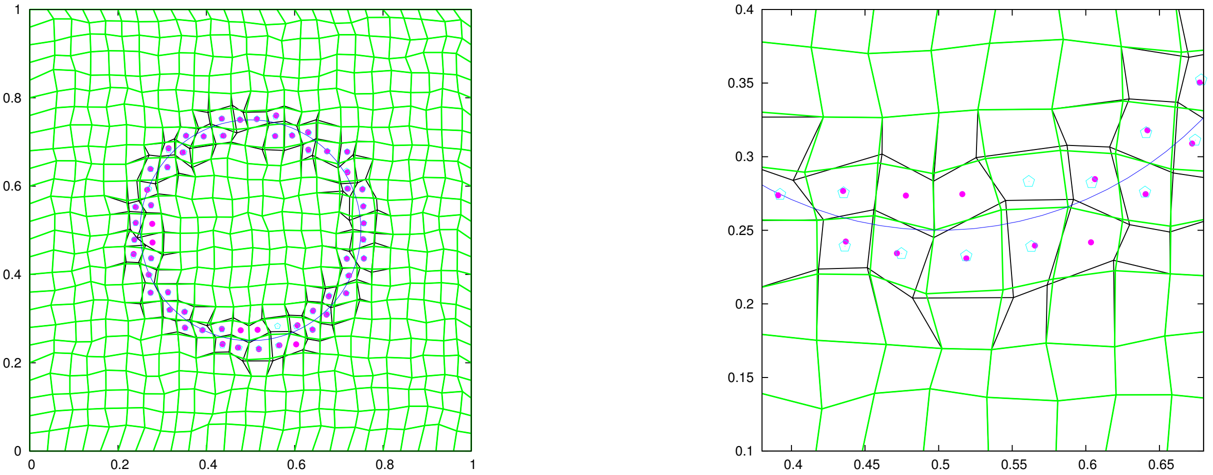
In the second step, we start with the intermediate grid where we know all data and move only mixed points. Again, we can use any rezoning algorithm keeping pure points fixed in the position that they had been moved to in the first step. We assume that the rezoning algorithm creates a valid grid. The second step creates the final rezoned (new) grid, Fig. 11, which will be used at the beginning of next Lagrangian step. Remapping from the intermediate grid to the new grid is performed using an intersection based method. We now explain the details of this step.

Clearly, in the second step, a mixed cell can exchange data only with other mixed cells or with pure cells which share an edge or a vertex with this mixed cell. We note that there are degenerate situations when a mixed point is shared by pure cells that do not all contain the same material. In this situation, pure cells of different materials are exchanging mass and all of them become mixed. This observation leads to the notion of *buffer* cells. Such buffer cells are cells that can be involved in data exchange in the second step. Buffer cells are cells which have at least one mixed point as their vertex. In Fig. 12 we depict mixed and buffer cells in the old grid.

If the MFIB method is used in the second step of hybrid remapping, only fluxes need to be computed that are related to edges for which at least one end point is a mixed point, and for fluxes related to mixed points. We denote such edges as *mixed* edges in contrast to *pure* edges, for which both end points are pure points. If the MAIB method is used in the second step of hybrid remapping, it only needs to be performed for mixed and buffer cells.

Before performing the second step we need to reconstruct the density function in buffer cells using data obtained from buffer cells and its pure neighbors that were the result of the first step SRB remapping.

[5] Strictly speaking, this is not exactly true, because densities and geometry of surrounding pure cells has been changed. If one decides to perform interface reconstruction using the updated grid and geometry using the VOF method, the interface in mixed cells can change slightly. If one uses MOF interface reconstruction instead, then the interface will not change because MOF does not use information from neighboring cells. In this work we use MOF for interface reconstruction and therefore we do not need this additional step.



a) b)

Figure 11: Overlap of intermediate grid (solid black line) and final new grid (solid green line). Left panel entire grid: mixed cells on old grid are marked by magenta solid circles; mixed cells on new grid are marked by cyan pentagons. Right panel - fragment, interface is also shown. Some cells which were mixed on old grid now is pure and vise versa.

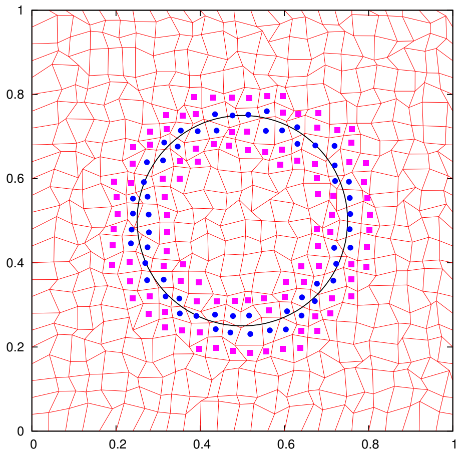


Figure 12: Mixed and buffer cells. Mixed cells are marked by blue solid circles, and buffer cells are marked by magenta solid squares. Mixed cells are not affected in the first step of hybrid remapping. Data in buffer cells is updated in the first step of hybrid remapping. The second step of remapping requires the reconstruction of functions in buffer cells, because these cells will exchange information with mixed cells and/or between each other.

Before we summarize the main stages of our two step hybrid rezoning/remapping algorithm for the multimaterial case, we recall the definitions for the various types of mesh entities.

- point types:
  - pure - the union of its adjacent cells contains one material
  - mixed - the union of its adjacent cells contains more than one material
- edge types:
  - pure - both endpoints are pure points
  - mixed - one of its endpoints is a mixed point
- cell types:
  - pure - contains one material
  - mixed - contains more than one material
  - buffer - has at least one mixed point as a vertex

We continue by outlining the main stages of our two step hybrid rezoning/remapping algorithm for the multimaterial case:

- Preprocessing step. Mark pure and mixed points, buffer cells, and edges for which fluxes must be computed. In the simplest implementation this requires at least one sweep over the entire grid and therein nested loops over neighbors of cells and points.
- Reconstruct the function over pure cells. Here, the computation of the gradient must be done without taking into account contributions from mixed cells as they contain sub-polygons. The gradient used for SRB will be zero in mixed cells.
- Move pure points.
- Remap using a swept-region-based method. This is a loop over *pure* edges and the corresponding exchange of mass between cells sharing this edge.
- Reconstruct the function over buffer cells and pure sub-polygons.
- Move mixed points.
- Remap using an intersection-based multimaterial method. In the case of MFIB this is a loop over mixed edges and the corresponding exchange of information between cells sharing a mixed edge. This involves all work which is done in MFIB per edge, such as intersection of a new cell with pure sub-polygons and so on, see section 4.3. In particular, this requires the remapping of centroids if MOF interface reconstruction is used.

We denote this new two-step hybrid multimaterial algorithm by MHYB.

Note that it is critical that pure cells are processed first. If mixed cells were processed first, a new mixed cell might be created in the intermediate mesh.

5.4. Relative Cost of Two-Step Hybrid Algorithm

We now give a very rough estimate of the computational cost of the hybrid method and give a comparison with the MFIB method on the example with two materials and a logically rectangular grid. We start with the MFIB algorithm applied to the entire grid. As we have mentioned before, we need to estimate the total number of cells, edges and points in the grid, which is composed of pure cells and pure sub-polygons. Then an estimate similar to (26) can be used. We denote the number of pure, mixed and buffer cells by $N_{cell,pure}$, $N_{cell,mixed}$ and $N_{cell,buffer}$, respectively; and the total number of cells by $N_{cell,total} = N_{cell,pure} + N_{cell,mixed}$. For a logically rectangular grid, the total number of cells is $N_{cell,total} = N * M$, the total number of edges is $N_{edge,total} = N*(M+1)+M*(N+1)$, and the total number of points is $N_{point,total} = (N+1)*(M+1)$. In the case of two materials, in each mixed cell we have two pure sub-polygons. Hence, the total number of cells in the *composed* grid is $N_{cell,total} + N_{cell,mixed}$. In case of a quad grid and two materials pure sub-polygons may be triangles, quads or pentagons. For the purpose of counting additional edges and vertices we can assume that all of them are quads. Under this assumption we add two new vertices and four new edges per mixed cell. Under these assumptions the cost of MFIB is

$$\begin{aligned} C_{MFIB} &= (N_{cell,total} + N_{cell,mixed}) * C_{rec} + \\ &2 * \left[(2 * N_{point,total} + 2 * N_{cell,mixed}) + (N_{edge,total} + 4 * N_{cell,mixed})\right] \left[C_{int} + C_{integ}\right]. \end{aligned} \tag{40}$$

Now, if we assume that $N_{cell,mixed} = C_{mixed}\sqrt{N_{cell,total}}$ then for fine grids we obtain the following estimate

$$C_{MFIB} \sim (N*M+C_{mixed}*\sqrt{N*M})*C_{rec}+2*[3*N*M+6*C_{mixed}*\sqrt{N*M}]\left[C_{int}+C_{integ}\right]. \tag{41}$$

We now estimate the cost of the hybrid algorithm for the same two-material case. We can ignore the cost of rezoning because rezoning must be done for MFIB as well and is not accounted for in estimate (41)[6]. Clearly, the new part of the algorithm is preprocessing; we denote its total cost as $\mathcal{C}_{prep}$ - this is the additional cost in comparison with the cost of MFIB. It is not clear that $\mathcal{C}_{prep}$ can be represented as $N_{cell,total} * C_{prep}$, where C_{prep} is constant and does not depend on grid size. However, in our implementation we did not observe such a dependence. Another additional cost is related to function reconstruction in buffer cells twice. Now, the total cost of MHYB remapping is

$$\begin{aligned} C_{MHYB} &= \mathcal{C}_{prep} + (N_{cell,total} + N_{cell,mixed?} + N_{cell,buffer}) * C_{rec} + N_{edge,pure} * C_{integ} + \\ &2 * \left[(4 * N_{cell,mixed}) + 2 * N_{cell,buffer})\right] \left[C_{int} + C_{integ}\right]. \end{aligned} \tag{42}$$

where the first term on the RHS of (42) is the total cost of reconstruction; the second term on the RHS of (42) is the approximate cost of SRB remap (excluding reconstruction) performed in the first stage of hybrid remap, and the last term on the RHS is the approximate cost of MFIB remap (excluding reconstruction) in the second stage of hybrid remap.

Now we assume that $N_{cell,buffer} \sim 2*N_{cell,mixed} = 2*C_{mixed}*\sqrt{N*M}$, and that the number of pure edges is proportional to the number of pure cells (which is the total number of cells minus the number of mixed) - $N_{edge,pure} \sim N*M - C_{mixed}*\sqrt{N*M}$. Under these assumptions, we obtain

[6] Again, this is not exactly true because in the hybrid algorithm we need to check if a point is pure or mixed before moving it. This has its own cost, which we ignore for this rough estimate.

the following estimate for a fine grid

$$C_{MHYB} \sim \mathcal{C}_{prep} + (N * M + 3 * C_{mixed} * \sqrt{M * N}) * C_{rec} + (N * M - C_{mixed} * \sqrt{N * M}) * C_{integ} + 16 * C_{mixed} * \sqrt{N * M} * [C_{int} + C_{integ}] \,. \tag{43}$$

If the grid is fine enough such that we can ignore terms of order $\sqrt{N * M}$ then we get

$$\begin{aligned} C_{MFIB} &\sim N * M * (C_{rec} + 6 * [C_{int} + C_{integ}]) \,, \\ C_{MHYB} &\sim \mathcal{C}_{prep} + N * M * (C_{rec} + C_{integ}) \,. \end{aligned} \tag{44}$$

This estimate suggests that the relative efficiency of the two methods depends mostly on the relative cost of intersections and the cost of the preprocessing stage. We believe that a more detailed analysis only makes sense for a particular code and a particular implementation of the algorithms. In the next section, we give comparisons based on wall clock time for two specific implementations in two different codes, respectively.

6. Numerical Experiments

In this section we present results obtained using the algorithms implemented in two multimaterial ALE codes.

6.1. Brief Description of Two Multimaterial Codes

The *first code*, [20, 18], uses a staggered discretization on a logically rectangular grid consisting of general quadrilateral cells. In a staggered discretization, density, internal energy and pressure are cell-centered, and the velocity vector is defined at grid points. Staggered discretizations require special algorithms for remapping nodal quantities, (see for example, [4, 22]). We will not describe such algorithms here and refer the interested reader to cited papers and the review paper [3]. For the purpose of this paper, it is only important that the remapping of nodal quantities may require some additional work which can increase the cost of overall remapping. In this code, we were originally using MFIB remapping for multimaterial problems and SRB remapping for single material problems. In the new version of the code, we have implemented the new hybrid remapping described in this paper, where in the second multimaterial step we use MFIB remapping. In the rest of the paper we will call this code ST - for staggered. In the ST code, the old grid may have non-convex cells as a result of the Lagrangian step.

The ST code employs MOF, [1, 7], as interface reconstruction method. For each mixed cell, the MOF package returns a set of triangles which represent a particular material. Even if a material can be represented by one polygon inside a mixed cell, the MOF package subdivides it into triangles. This is done to allow unified processing of the materials even in the case when a material is represented by several disjoint pieces.

When a new cell is intersected with a pure old cell (which maybe non-convex) we use an algorithm for the intersection of two quads. The algorithm is based on half-plane intersections (half-planes are defined by the edges of the new cell) with the old cell; it is an extremely robust algorithm that for close to parallel line segments is bisection-based, [19]. In case of an intersection of a new cell with a multimaterial cell we intersect the new cell with each triangle representing the material using the same algorithm as quad-quad intersection. We note that in any case, the new cell is always represented as a quad and it is not subdivided into triangles.

The *second code* uses a cell-centered discretization on a general polygonal grid, [9]. In a cell-centered discretization, all quantities are cell-centered and a special algorithm for movement of the nodes is used. Therefore, all quantities are remapped in a unified way using only cells. In this code we were originally using MAIB remapping for multimaterial problems and SRB remapping for single material problems. In the new version of the code we have implemented the new hybrid remapping described in this paper, where in the second multimaterial step we use MAIB remapping. In the rest of the paper we will refer to this code as CC - for cell-centered. In the CC code, we also use the MOF method for interface reconstruction. The CC code differs from the ST code in the way intersections are performed. In this code not only pure sub-polygons representing materials in mixed cells are subdivided into triangles, but all pure cells of the old grid are subdivided into triangles. Cells of the new grid are also subdivided into triangles and all intersection-based remapping processes are reduced to triangle-triangle intersections, where the first triangle represents a cell of the new grid and the second triangle represents a pure or mixed cell of the old grid. Details of this approach can be found in [9]. For the purpose of this paper, it is important to note that this process is more expensive than the one used in the ST code, since it requires many more intersections. Note also that, in 3D, cells may become non-convex as their faces become non-planar. Thus, it will be necessary to perform subdivisions of the cells into tetrahedra to perform intersections. Therefore, the timings for the CC code can give an idea about the relative cost of the MAIB remapping method in the entire domain and about hybrid remapping in 3D.

The choice of these two codes for performing numerical experiments allows us to demonstrate different scenarios for our new multimaterial hybrid remapping strategy in multimaterial ALE codes as well as give an idea about the relative cost of the different methods.

6.2. Stand-alone Remapping

In this section we compare different remapping methods using a stand-alone cyclic remapping problem. To that end, we define masses of the materials on the initial grid according to a specified density function, then do several remapping steps changing the grid and using results of the previous remapping step as initial data for the next remapping step. After several remapping steps we return to the original grid and compare remapped with the original masses. This approach for testing remapping was introduced in [25].

The computational domain is a unit square. In this domain we have a circle of radius 0.25 with its center at $(0.5, 0.5)$. For a single material problem, this circle is used to define a discontinuous density function. For a multimaterial problem with two materials, this circle contains a material which is different from the material in the rest of computational domain. In Fig. 13 a) we depict this circle and a uniform grid.

We define the initial distribution of density as follows

$$f(x,y) = \begin{cases} 1 + e^{10\sqrt{(x-1/2)^2+(y-1/2)^2}} & \text{for } \sqrt{(x-1/2)^2+(y-1/2)^2} \le 1/4\,, \\ 1 + e^{6\left(\sqrt{(x-1/2)^2+(y-1/2)^2}-1/4\right)} & \text{in the rest of computational domain.} \end{cases} \tag{45}$$

This density distribution is used both for the single material and for the two-material problem.

For the multimaterial problem the interface between materials inside a mixed cell is represented by a segment of a straight line. This line segment is determined using a PLIC (piece-wise linear interface construction) method. In this paper, we use the MOF method, which requires knowledge of volume fraction and centroid of each material. Interface reconstruction results in two pure

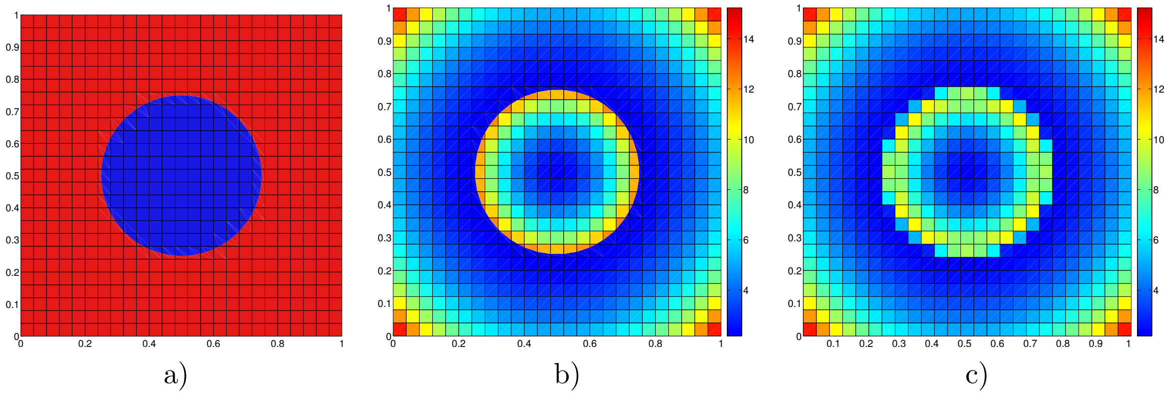


Figure 13: Cyclic remapping problem: a) Initial square grid. b) Initial density distribution for multimaterial problem; c) Initial density distribution for single material problem.

material sub-polygons in each mixed cell. Then one computes mass and mean density in each pure material sub-polygon using the corresponding density function (45). The color map for mean values of density for multimaterial problem is presented in Fig. 13 b).

For the single material problem, the entire computational domain is occupied by one material and the circle is only used to define a discontinuous density. In this case, formally, there are no mixed cells. However, in each cell which is intersected by the circle we define one mass by summing masses of ”materials” as described for the multimaterial case. The mean density for such cells equals mass divided by total volume. The color map for mean values of density for the single material problem is presented in Fig. 13 c).

Here, we first move all vertices using the following formulas

$$X^{New} = X^{Ini} + a_0 \Delta x \cos(2\ \pi\ \text{rand})\,, \quad \mathrm{Y}^{\mathrm{New}} = \mathrm{Y}^{\mathrm{Ini}} + \mathrm{a}_0 \Delta \mathrm{y} \sin(2\ \pi\ \mathrm{rand})\,, \tag{46}$$

where $a_0 = 0.2$ for structured meshes and $a_0 = 0.1$ for polygonal meshes; $\Delta x = \Delta y = \dfrac{1}{N}$ (where N is a parameter that defines the number of cells in one dimension) and $0 < \text{rand} < 1$ is a random number to create perturbed grid. After the mesh was perturbed we then move points back to their initial positions.

The error of remapping is measured using the following relative norms. For single material problems

$$L_{1\,m} = \frac{\sum_{\forall c} |m_c^0 - m_c|}{\sum_{\forall c} m_c^0}\,, \tag{47}$$

where m_c^0 is the initial mass of cell c and m_c is the mass of the same cell on the initial grid after two remapping steps (initial to perturbed grid and back to initial grid).

For the multimaterial problem, we combine errors for each material in one error as follows

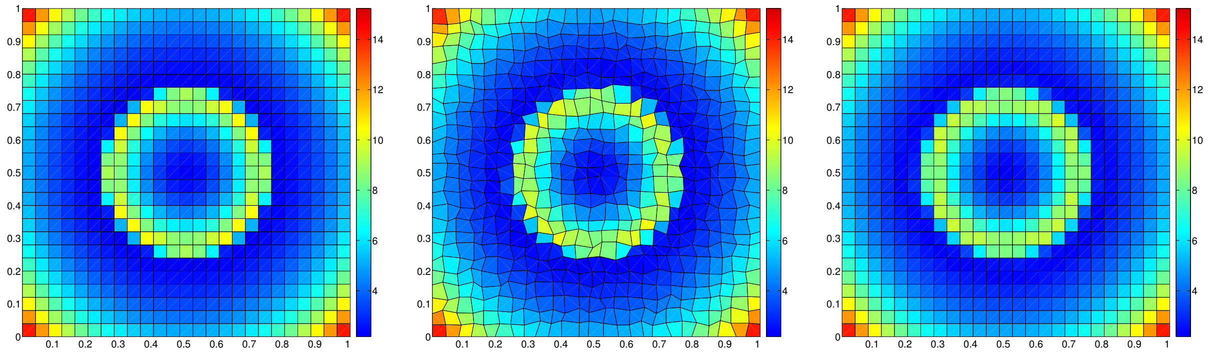
$$L_{1\,m}^{\text{mat}} = \frac{\sum_{\forall c} \sum_{\forall k} |m_{c,k}^0 - m_{c,k}|}{\sum_{\forall c} \sum_{\forall k} m_{c,k}^0}\,, \tag{48}$$

where k is the material index and $m_{c,k}^0\,, m_{c,k}$ are the initial and final mass of material k in cell c.

To investigate convergence and also compare CPU (Central Processing Unit) time required by different methods we perform cyclic remapping as described before on a series of refined grids.

In the first experiment, we use initial uniform grids of $N \times N$ cells, $N = 25$, 50, 100, 200, 400; and use grid movement defined by equation (46).

First, we present the result of cyclic remapping using cell-centered remapping from the ST code. For the single material case, we present the density color map on the initial grid, on the perturbed grid (after one remapping) and on the initial grid after the second remapping, Fig. 14.



SM, density, init.

SM, density, pert.

SM, density, final

Figure 14: Density distribution for initial, perturbed, and final grid of SM static remap test for double-exponential function using algorithms from ST code.

In Table 1, we present errors and timings for the single material cyclic remapping test. Time measurements are all performed on the same particular computer and it only makes sense to compare relative CPU time needed for different methods.

resolution	swept	intersection	hybrid
25×25, $L_{1\,m}$	$0.16 \cdot 10^{-1}$	$0.18 \cdot 10^{-1}$	$0.16 \cdot 10^{-1}$
25×25, T[s]	$2.20 \cdot 10^{-2}$	$7.90 \cdot 10^{-2}$	$2.90 \cdot 10^{-2}$
50×50, $L_{1\,m}$	$0.74 \cdot 10^{-2}$	$0.83 \cdot 10^{-2}$	$0.74 \cdot 10^{-2}$
50×50, T[s]	$8.30 \cdot 10^{-2}$	$2.79 \cdot 10^{-1}$	$1.23 \cdot 10^{-1}$
100×100, $L_{1\,m}$	$0.44 \cdot 10^{-2}$	$0.48 \cdot 10^{-2}$	$0.44 \cdot 10^{-2}$
100×100, T[s]	$3.01 \cdot 10^{-1}$	$1.03 \cdot 10^{0}$	$3.74 \cdot 10^{-1}$
200×200, $L_{1\,m}$	$0.20 \cdot 10^{-2}$	$0.22 \cdot 10^{-2}$	$0.20 \cdot 10^{-2}$
200×200, T[s]	$1.12 \cdot 10^{0}$	$4.14 \cdot 10^{0}$	$1.44 \cdot 10^{0}$
400×400, $L_{1\,m}$	$0.10 \cdot 10^{-2}$	$0.11 \cdot 10^{-2}$	$0.10 \cdot 10^{-2}$
400×400, T[s]	$4.48 \cdot 10^{0}$	$1.66 \cdot 10^{1}$	$5.72 \cdot 10^{0}$

Table 1: L_1 mass errors and simulation times, T[s] (in seconds), for single-material case using algorithms from ST code.

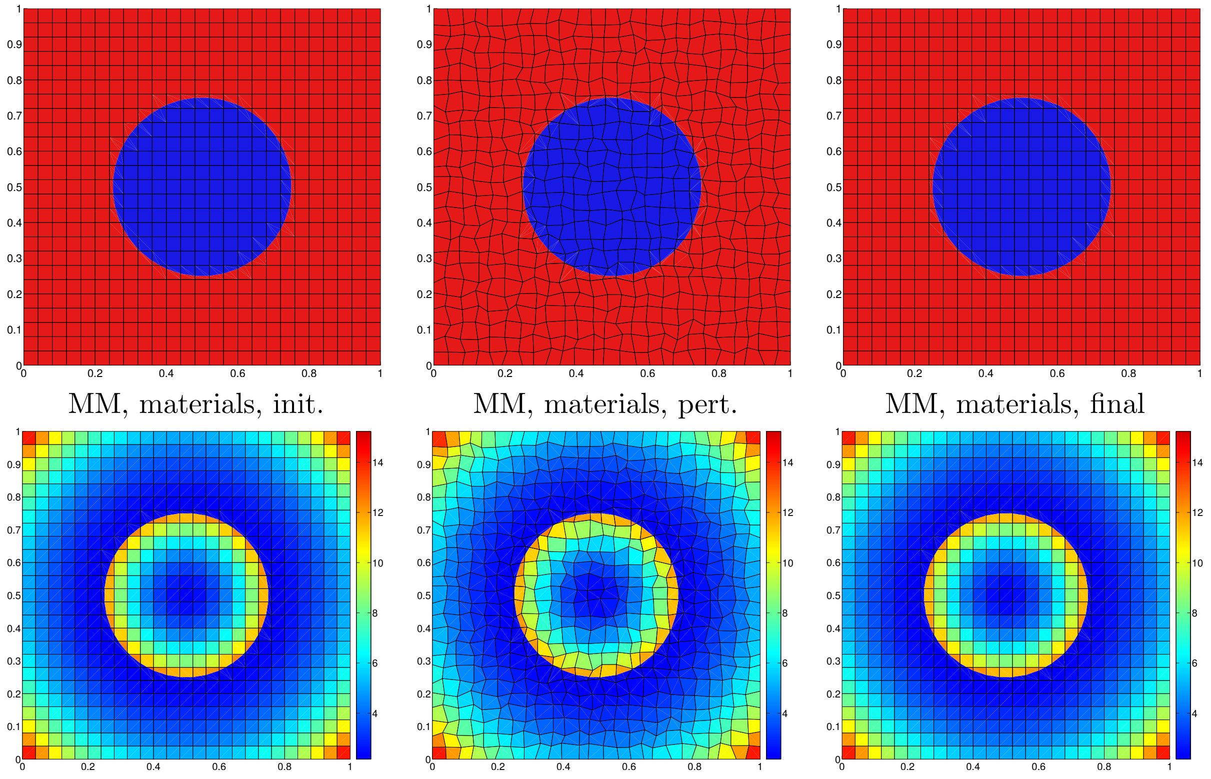
First, Table 1 shows that all methods demonstrate approximately first-order convergence, which is what is expected when remapping a discontinuous function. Second, the accuracy of the hybrid method almost equals the accuracy of either the SRB or FIB methods. One might expect that intersection based methods should be more accurate than swept-region based methods. However, in this example this is not the case. This can be attributed to the random perturbation of the grid;

in all cases the errors are very close. For the purpose of this paper it is important to note that the hybrid method is as accurate as the intersection based method.

Now we consider CPU time. For all methods CPU time increases approximately between three and four times with increasing resolution - this indicates that CPU time per cell is approximately a constant.

Now consider the finest resolution 400×400 grid, which can be considered close to the case when asymptotic estimates presented in previous sections are valid. For this example and this resolution SRB remapping is ~ 3.7 times cheaper than FIB remapping - which gives an indication as to the maximum possible gain in CPU time. In the case of a single material the new hybrid remapping method produces results, which coincide with SRB remapping because there are no mixed cells and swept-region remapping is performed for the entire domain. However, the hybrid method requires preprocessing to mark nodes and edges which takes additional time. The new hybrid remapping method is only ~ 2.9 times cheaper than FIB remapping. This indicates that some gain of the hybrid method over intersection-based remapping is lost due to preprocessing. In this particular case, the overhead due to preprocessing is about 25% in comparison with SRB remapping.

We now consider a multimaterial example, which uses the same density function as in the single material case. However, we now have two materials - one is inside and another outside the circle. In the left panel in Fig. 15, we present materials (result of interface reconstruction) and color map for density for the initial grid; in the central panel the same information is depicted for the perturbed grid; and the right panel again depicts the same information for the final grid.



MM, materials, init.

MM, materials, pert.

MM, materials, final

MM, density, init.

MM, density, pert.

MM, density, final

Figure 15: Material and density distribution for initial, perturbed, and final grid using algorithms for ST code.

In Table 2, we present errors and timings for multimaterial cycling remapping starting with a square grid on a sequence of refined grids.

resolution	intersection	hybrid
25×25, $L_{1\,m}^{\mathrm{mat}}$	$0.19 \cdot 10^{-2}$	$0.21 \cdot 10^{-2}$
25×25, T[s]	$1.84 \cdot 10^{-1}$	$1.57 \cdot 10^{-1}$
50×50, $L_{1\,m}^{\mathrm{mat}}$	$0.39 \cdot 10^{-3}$	$0.40 \cdot 10^{-3}$
50×50, T[s]	$4.93 \cdot 10^{-1}$	$3.84 \cdot 10^{-1}$
100×100, $L_{1\,m}^{\mathrm{mat}}$	$0.88 \cdot 10^{-4}$	$0.87 \cdot 10^{-4}$
100×100, T[s]	$1.56 \cdot 10^{0}$	$9.73 \cdot 10^{-1}$
200×200, $L_{1\,m}^{\mathrm{mat}}$	$0.28 \cdot 10^{-4}$	$0.27 \cdot 10^{-4}$
200×200, T[s]	$5.64 \cdot 10^{0}$	$3.06 \cdot 10^{0}$
400×400, $L_{1\,m}^{\mathrm{mat}}$	$0.96 \cdot 10^{-5}$	$0.97 \cdot 10^{-5}$
400×400, T[s]	$2.09 \cdot 10^{1}$	$1.06 \cdot 10^{1}$

Table 2: L_1 mass errors and simulation times, T[s] (in seconds), for the multimaterial case using algorithms from the ST code.

This table demonstrates that for this example, the accuracy of MFIB and hybrid methods is almost equal and the convergence for both methods is between first- and second-order, but closer to second-order. This can be explained as follows. As a result of interface reconstruction, we effectively construct a finer grid which increases resolution near the interface and in our case near the line where the function has a discontinuity. Also the function reconstruction for the multimaterial case uses a special procedure to reconstruct the slope in pure sub-polygons in mixed cells, [20, 18], which in contrast to many other codes leads to a non-zero slope because it uses information from pure sub-polygons containing the same material from surrounding cells (mixed and pure).

When comparing CPU times for MFIB and hybrid methods, we see that their ratio increases from 1.17 to 1.97 with grid refinement, which suggests that for a finer grid, the hybrid method is about twice, or perhaps even more than twice as efficient as MFIB remapping.

For both methods, CPU time increases approximately between 2.5 and 3.7 times with increasing resolution - which indicates that CPU time per cell is approximately constant.

Now we present results for the same problem using algorithms employed in the CC code. We remind the reader that the main differences to the ST code are that the CC code uses the MAIB method instead of MFIB and that intersections are done by subdividing all polygons into triangles.

Here we present only tables. For the single material case, we present results in Table 3, which is analogous to Table 1.

First, we conclude that accuracy and convergence rate are the same as for the ST code. Theoretically they should be perfectly equal, since the ST and CC codes are just different implementations of the same formulas. However, the codes are using different random number generators, which produces slightly different grid motion. The discrepancy in timings between the CC and ST codes for the SRB method is due to the fact that different computers and different compilers were used.

We now consider the timings for the intersection-based algorithm. From a comparison with results in Table 1 we conclude that the intersection-based algorithm implemented in the CC code is 16 to 28 times more expensive (depending on resolution) than the intersection-based algorithm implemented in the ST code. This can be explained by the fact that each quad cell of the old and new grid is subdivided into triangles and we need to intersect each triangle representing the new

resolution	swept	intersection	hybrid
25×25, $L_{1\,m}^{\rm mat}$	$0.13\cdot 10^{-1}$	$0.17\cdot 10^{-1}$	$0.13\cdot 10^{-1}$
25×25, T[s]	$1.0\cdot 10^{-2}$	$1.22\cdot 10^{0}$	$3.0\cdot 10^{-2}$
50×50, $L_{1\,m}^{\rm mat}$	$0.68\cdot 10^{-2}$	$0.85\cdot 10^{-2}$	$0.68\cdot 10^{-2}$
50×50, T[s]	$4.0\cdot 10^{-2}$	$5.0\cdot 10^{0}$	$1.3\cdot 10^{-1}$
100×100, $L_{1\,m}^{\rm mat}$	$0.37\cdot 10^{-2}$	$0.46\cdot 10^{-2}$	$0.37\cdot 10^{-2}$
100×100, T[s]	$2.1\cdot 10^{-1}$	$2.0\cdot 10^{1}$	$5.5\cdot 10^{-1}$
200×200, $L_{1\,m}^{\rm mat}$	$0.17\cdot 10^{-2}$	$0.22\cdot 10^{-2}$	$0.17\cdot 10^{-2}$
200×200, T[s]	$9.1\cdot 10^{-1}$	$8.02\cdot 10^{1}$	$2.26\cdot 10^{0}$
400×400, $L_{1\,m}^{\rm mat}$	$0.81\cdot 10^{-3}$	$0.1\cdot 10^{-2}$	$0.81\cdot 10^{-3}$
400×400, T[s]	$3.65\cdot 10^{0}$	$3.25\cdot 10^{2}$	$9.45\cdot 10^{0}$

Table 3: L_1 mass errors and simulation times, T[s] (in seconds), for single-material case using algorithms from CC code.

cell with triangles representing cells of the old grid in the neighborhood of the old cell. This by itself can account for factor of four difference in CPU time. Additionally, the AIB method is more expensive then the FIB method. There are also differences to the ST code's triangle intersection algorithm. Another very important difference is that CC code is an unstructured code and memory access to geometric data from a neighboring cell may take much longer. Our goal here is not to analyze why the difference between two implementations of intersection algorithms is so dramatic, but to demonstrate what can happen in real relatively simple ALE codes.

For finer resolution, the cost of the hybrid method using the CC code is a little bit higher (because of different overhead, which is about 100% in this case), but comparable with the hybrid method using the ST algorithms. This is because for a finer grid, the cost is dominated by the SRB algorithm, for which timing is almost the same for both codes.

The relative efficiency of the hybrid method in comparison with the intersection-based method is much higher for the CC code than for the ST code. For the ST code it is around two, and for the CC code it is about 40. This is because the CC code's intersection-based method is so expensive.

We now consider the multimaterial case. The results are presented in Table 4. First, the convergence rates for both intersection-based and hybrid methods are about the same as when algorithms from the ST ALE code are used, because it is essentially an equivalent implementation.

In the multimaterial case for the finest grid, the hybrid method is about twenty times more efficient than the intersection-based method. Recall that when using the ST code, the hybrid method was only twice as efficient as the intersection-based method.

It is interesting to observe that for intersection-based method CPU time increases proportionally to the number of cells, in particular, about four times for two consecutively refined grids. For the hybrid method this increase is only two times. Our explanation for this behavior is that the CC code's intersection algorithm is very expensive, and dominates the overall cost. Additionally, in this particular example, the number of cells where intersections are needed is increasing roughly by a factor of two under regular refinement. Eventually for fine enough resolution the cost of swept remapping will dominate and CPU time will increase four times with doubling of the resolution in each coordinate direction.

The CC code is an unstructured code which can handle arbitrary polygons. We demonstrate its performance on the cyclic remapping example described at the beginning of this section, but

resolution	intersection	hybrid
25×25, $L_{1\,m}^{\mathrm{mat}}$	$0.18 \cdot 10^{-2}$	$0.17 \cdot 10^{-2}$
25×25, T[s]	$1.46 \cdot 10^{0}$	$5.3 \cdot 10^{-1}$
50×50, $L_{1\,m}^{\mathrm{mat}}$	$0.32 \cdot 10^{-3}$	$0.33 \cdot 10^{-3}$
50×50, T[s]	$5.45 \cdot 10^{0}$	$1.15 \cdot 10^{0}$
100×100, $L_{1\,m}^{\mathrm{mat}}$	$0.77 \cdot 10^{-4}$	$0.81 \cdot 10^{-4}$
100×100, T[s]	$2.09 \cdot 10^{1}$	$2.52 \cdot 10^{0}$
200×200, $L_{1\,m}^{\mathrm{mat}}$	$0.17 \cdot 10^{-4}$	$0.18 \cdot 10^{-4}$
200×200, T[s]	$8.24 \cdot 10^{1}$	$6.27 \cdot 10^{0}$
400×400, $L_{1\,m}^{\mathrm{mat}}$	$0.42 \cdot 10^{-5}$	$0.45 \cdot 10^{-5}$
400×400, T[s]	$3.27 \cdot 10^{2}$	$1.79 \cdot 10^{1}$

Table 4: L_1 mass errors and simulation times, T[s] (in seconds), for multimaterial case using algorithms from CC code.

using an initial polygonal grid.

The initial polygonal grid is presented in Fig. 16 a). This grid is a centroidal Voronoi grid, [6], where there are four generators at the corners of the computational domain and the same number of generators as number of squares in the previous quad grid examples. The result of interface reconstruction at the initial time is presented in Fig. 16 b).

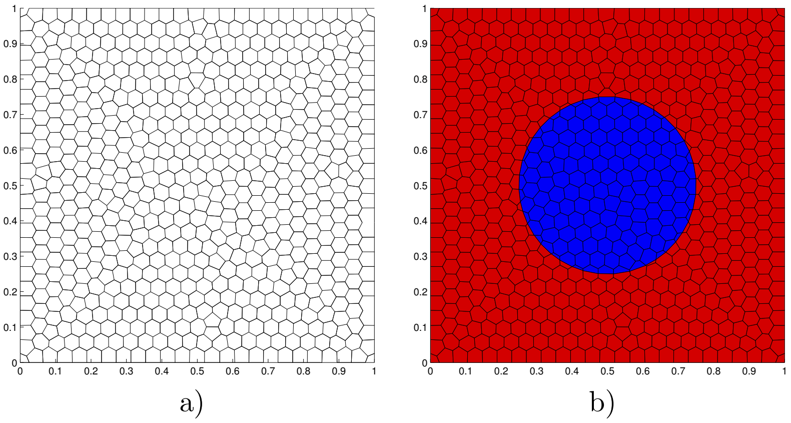
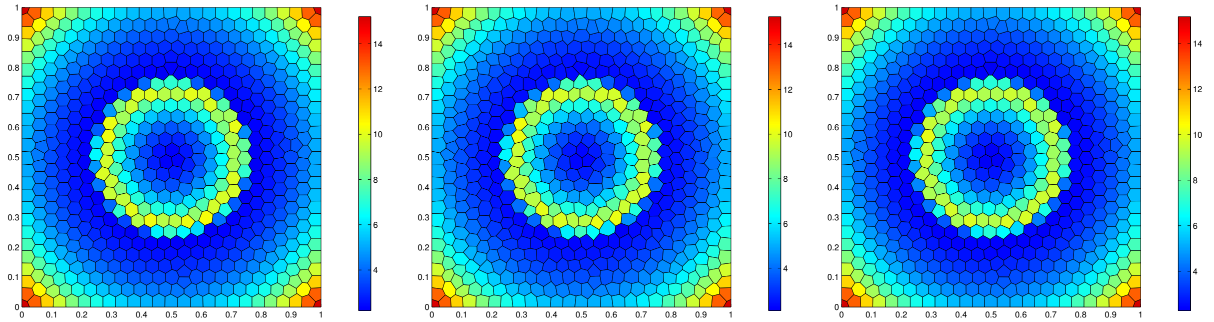


Figure 16: Initial time - CC code: a) Initial polygonal grid; b) Result of interface reconstruction.

In Fig. 17, we present the color map of density for the single material case. In Table 5, we present errors and timing for single material case. In general, for a polygonal grid all the methods are more expensive because cells have more edges, and must be subdivided into more triangles than for a quad grid. The qualitative conclusions are similar to the square grid case and we do not repeat them here.

In Fig. 18, we present results for interface reconstruction for initial, perturbed and final grid for the multimaterial case (top panel) and the color map for the density function on these grids. Errors and timings for the multimaterial case are presented in Table 6. Again, qualitative conclusions are the same as for the quad grid.



SM, density, init.   SM, density, pert.   SM, density, final

Figure 17: Density distribution for initial, perturbed, and final grid of SM remapping test - initial polygonal grid.

resolution	swept	intersection	hybrid
$25 \times 25,\ L_{1\,m}^{\mathrm{mat}}$	$0.78 \cdot 10^{-2}$	$0.83 \cdot 10^{-2}$	$0.78 \cdot 10^{-2}$
$25 \times 25,\ T[\mathrm{s}]$	$1.99 \cdot 10^{-2}$	$3.72 \cdot 10^{0}$	$4.00 \cdot 10^{-2}$
$50 \times 50,\ L_{1\,m}^{\mathrm{mat}}$	$0.44 \cdot 10^{-2}$	$0.50 \cdot 10^{-2}$	$0.44 \cdot 10^{-2}$
$50 \times 50,\ T[\mathrm{s}]$	$8.00 \cdot 10^{-2}$	$1.54 \cdot 10^{1}$	$1.7 \cdot 10^{-1}$
$100 \times 100,\ L_{1\,m}^{\mathrm{mat}}$	$0.18 \cdot 10^{-2}$	$0.21 \cdot 10^{-2}$	$0.18 \cdot 10^{-2}$
$100 \times 100,\ T[\mathrm{s}]$	$3.4 \cdot 10^{-1}$	$6.29 \cdot 10^{1}$	$7.4 \cdot 10^{-1}$
$200 \times 200,\ L_{1\,m}^{\mathrm{mat}}$	$0.94 \cdot 10^{-3}$	$0.11 \cdot 10^{-2}$	$0.94 \cdot 10^{-3}$
$200 \times 200,\ T[\mathrm{s}]$	$1.33 \cdot 10^{0}$	$2.56 \cdot 10^{2}$	$2.95 \cdot 10^{0}$

Table 5: Polygonal grid: L_1 mass errors and simulation times for single-material case - CC code

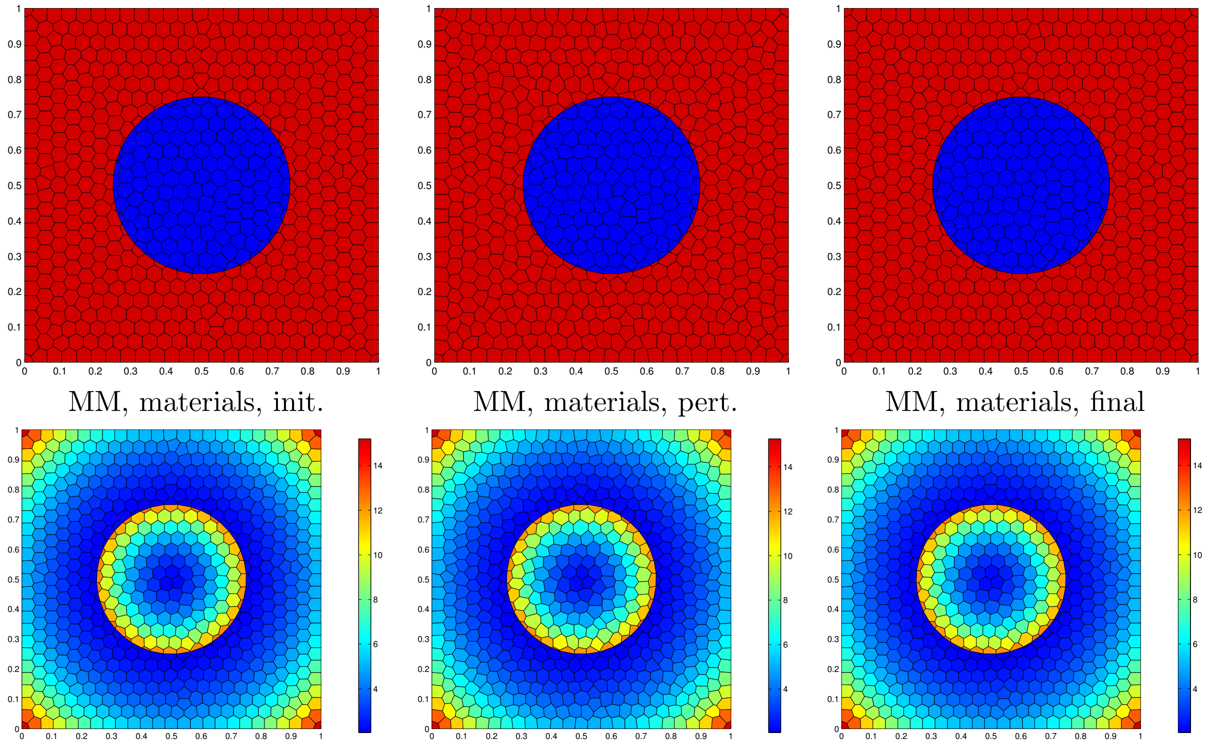
6.3. ALE Calculation of a Rayleigh-Taylor Instability

We are considering a Rayleigh-Taylor instability (RTI) problem for two immiscible ideal gases, one heavy and one light, with gravitational field directed vertically downward and with magnitude $g = 0.1$ The computational domain is defined by $(x, y) \in [0, 1/6] \times [0, 1]$. The initial density for the heavier gas is $\rho_1 = 2$, and $\rho_2 = 1$ for the lighter gas. Both gases have the same adiabatic constant $\gamma = 1.4$. The interface has been deliberately perturbed according to the formula $Y_i(x) = \frac{1}{2} + 10^{-2} \cos(6\pi x)$, see also Fig. 19. The initial pressure distribution is approximately hydrostatic and is defined as follows:

$$P_1(x,y) = 1 + \rho_1 \parallel \boldsymbol{g} \parallel (1-y), \text{ if y > Y}_\text{i}\text{(x)},$$
$$P_2(x,y) = 1 + \rho_1 \parallel \boldsymbol{g} \parallel (1-Y_i(x)) + \rho_2 \parallel \boldsymbol{g} \parallel (Y_i(x)-y), \text{ if y < Y}_\text{i}\text{(x)}.$$

It is well known that this configuration is unstable and as time progresses, the heavier gas will sink and the lighter gas will rise through the formation of bubbles and spikes. This problem does not involve any shock wave, but the vorticity is so high that pure Lagrangian schemes eventually fail, therefore we use ALE methods. The final time for this problem is $t = 10s$.

We model this problem using both the ST and CC ALE codes. Because this problem involves two gases with the same equation of state we can run it both in single material mode and multimaterial mode similar to the stand-alone remapping test presented in the previous section. This allows us to do a similar analysis of the numerical results.



MM, materials, init.

MM, materials, pert.

MM, materials, final

MM, density, init.

MM, density, inter.

MM, density, final

Figure 18: Material and density distribution for initial, perturbed, and final grid of MM cyclic remapping test - polygonal grid - CC code.

6.3.1. Single Material Calculations of RTI Problem

For the single material RTI problem, we can compare the swept-region method for the entire domain, the intersection-based method for the entire domain, and the hybrid method.

We start with results obtained from the ST code. We have chosen to use this code to run problem in Eulerian=Lagrange+Remap mode, that is, after each Lagrangian step we return back to the original grid. The initial grid is almost a rectangular grid (in right panel of Fig.20 we show such a grid for $17 \times 100 = 1700$ cells).

In the right panel in Fig. 20, we show the initial distribution of density. In the other panels, we show the distribution of density at the final time for different remapping methods: swept-region-based, intersection-based, and hybrid. The conclusion from these pictures is that they are practically identical.

We now consider the timings for this problem, which are presented in Table 7. Our fist observation is that the total time of calculation increases about ten times with increasing resolution, which indicates that there is no simple dependence of the total cost of ALE on the grid resolution. Our second observation is that for this example using the ST ALE code the cost of the hybrid method is almost the same as the cost of the swept-region-based method. So in the overall ALE calculation the relative cost of overhead related to the hybrid method is negligible. Finally, for the finest resolution, intersection-based remapping is about 27% more expensive than swept-region-based remapping. This may suggest that in this situation one may choose to use intersection-based remapping because in principle it should give more accurate results.

resolution	intersection	hybrid
25×25, $L_{1\,m}^{\text{mat}}$	$0.83 \cdot 10^{-3}$	$0.80 \cdot 10^{-3}$
25×25, T[s]	$4.04 \cdot 10^{0}$	$1.08 \cdot 10^{0}$
50×50, $L_{1\,m}^{\text{mat}}$	$0.15 \cdot 10^{-3}$	$0.15 \cdot 10^{-3}$
50×50, T[s]	$1.63 \cdot 10^{1}$	$2.44 \cdot 10^{0}$
100×100, $L_{1\,m}^{\text{mat}}$	$0.31 \cdot 10^{-4}$	$0.30 \cdot 10^{-4}$
100×100, T[s]	$6.49 \cdot 10^{1}$	$4.87 \cdot 10^{0}$
200×200, $L_{1\,m}^{\text{mat}}$	$0.78 \cdot 10^{-5}$	$0.70 \cdot 10^{-5}$
200×200, T[s]	$2.59 \cdot 10^{2}$	$1.16 \cdot 10^{1}$

Table 6: Polygonal grid: L_1 mass errors and simulation times for multimaterial case - CC code.

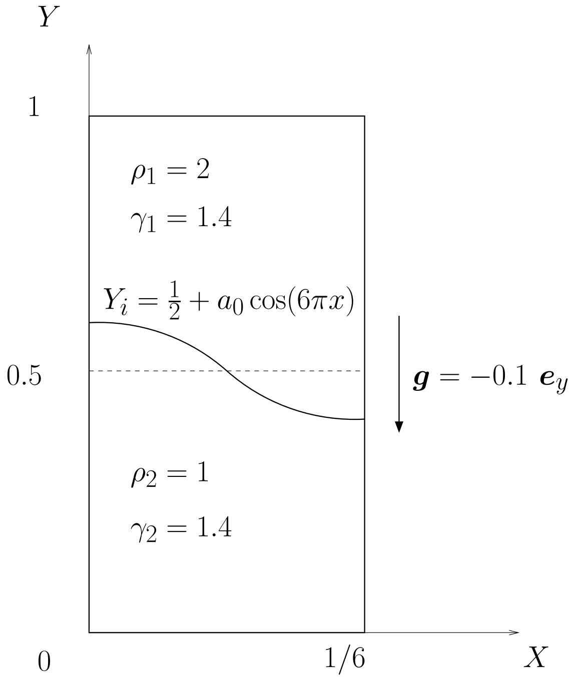
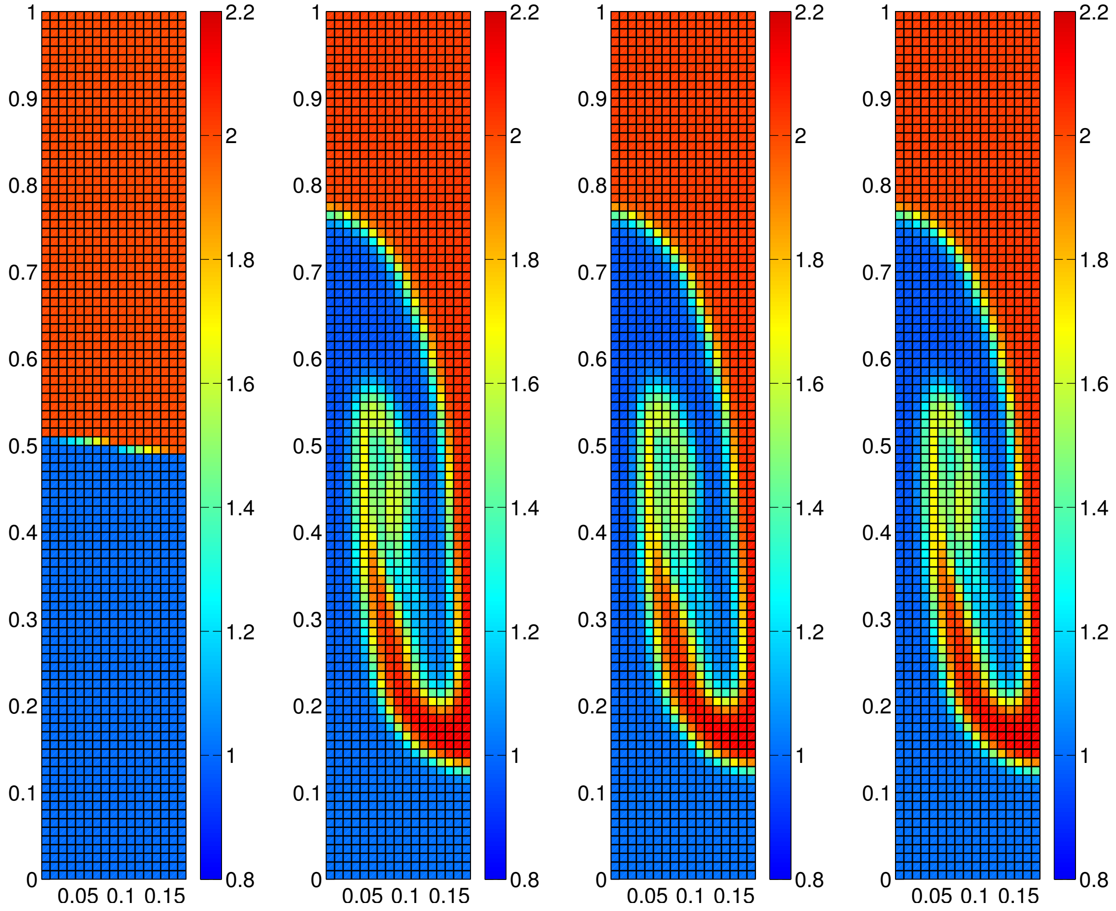

Y
1
$\rho_1 = 2$
$\gamma_1 = 1.4$
$Y_i = \frac{1}{2} + a_0 \cos(6\pi x)$
0.5
$\boldsymbol{g} = -0.1\ \boldsymbol{e}_y$
$\rho_2 = 1$
$\gamma_2 = 1.4$
0
1/6
X

Figure 19: Statement of Rayleigh-Taylor instability problem.

Now we present some results for the CC code. We chose to run this code using the initial polygonal grid shown in the right panel in Fig. 21. We run this problem in real ALE mode, where the grid is smoothed in each time step. In Fig. 21, we also present the grid and the color map for the density function at the final time for all remapping methods.

In the right panel in Fig. 21, we show the initial distribution of density. In the other panels we show the distribution of density at the final time for different remapping methods: swept-region-based, intersection-based, and hybrid. Our conclusion from these pictures is the same as for ST ALE method - they are practically identical to the naked eye.

For the CC ALE code, at a resolution comparable with 17×100 resolution on a rectangular grid, it takes $3.6 \cdot 10^3\ s$ to run this test using swept-region-based remapping, $3.83 \cdot 10^4\ s$ to run it using intersection based remapping and $4.54\ \cdot 10^3$ when the hybrid method is used. So the intersection-



init swept intersection hybrid

Figure 20: Single material Rayleigh-Taylor instability. Presented cell density in the initial and final time for all methods - ST code.

resolution	swept	intersection	hybrid
17×100	$1.42 \cdot 10^3$	$2.08 \cdot 10^3$	$1.51 \cdot 10^3$
33×200	$1.15 \cdot 10^4$	$1.65 \cdot 10^4$	$1.19 \cdot 10^4$
66×400	$9.80 \cdot 10^4$	$1.41 \cdot 10^5$	$1.01 \cdot 10^5$
133×800	$1.02 \cdot 10^6$	$1.40 \cdot 10^6$	$1.04 \cdot 10^6$

Table 7: Single material RTI problem - ST code timing results.

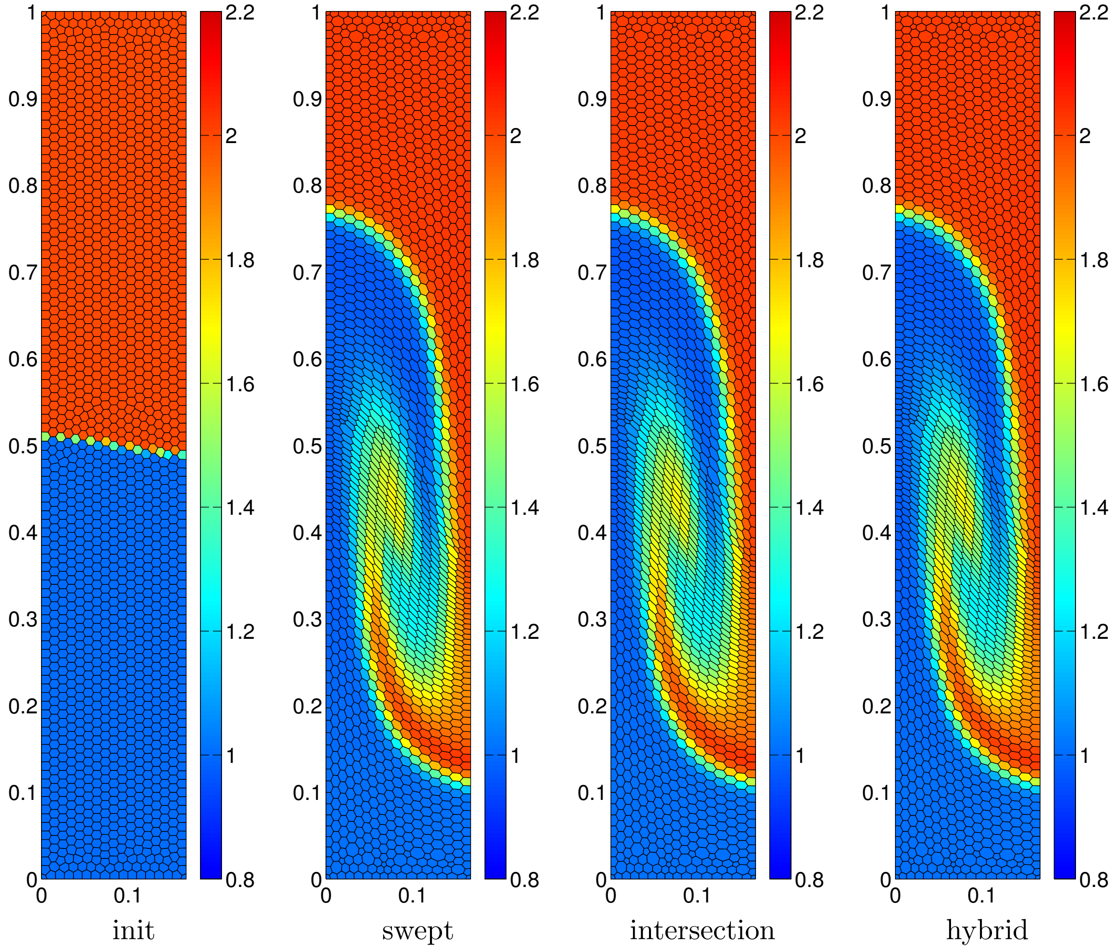


Figure 21: Single material Rayleigh-Taylor instability. Presented cell density in the initial and final time for all methods - CC ALE code.

resolution	intersection	hybrid
17×100	$3.47 \cdot 10^3$	$2.95 \cdot 10^3$
33×200	$2.68 \cdot 10^4$	$2.15 \cdot 10^4$
66×400	$2.37 \cdot 10^5$	$1.78 \cdot 10^5$
133×800	$2.21 \cdot 10^6$	$1.68 \cdot 10^6$

Table 8: Multimaterial RTI problem - ST code timing results.

based method is about eight times more expensive than the hybrid method and ten times more expensive than the swept-region-based method. This is a dramatic difference in comparison with the ST code, where the intersection-based method is only 1.37 times more expensive than the hybrid method and 1.4 times more expensive than the swept-region-based method. However, overall, the CC code is still much more expensive than the ST code.

6.3.2. Multimaterial Calculations of for the RTI Problem

For the multimaterial RTI problem, we only compare intersection-based and hybrid methods for both ST multimaterial ALE and CC multimaterial ALE codes.

Again, similar to the single material RTI problem, we run ST ALE in Eulerian=Lagrange+Remap mode.

In Fig. 22, we depict the color map for density in the corresponding material polygons (we do not show results of interface reconstruction because it looks almost identical to density color map) for the initial configuration and for the final time using intersection-based and hybrid remapping. From this we conclude that, for practical purposes, results for both methods are almost identical.

In Table 8, we present timings for different resolutions. We observe in Table 8 that for this problem, intersection-based remapping in the ST code is about 31% more expensive than hybrid remapping. We remind the reader that the timings in Table 8 are for the total calculation which includes the entire ALE algorithm.

Now, consider results obtained with the CC multimaterial ALE code, which we use in real ALE mode. In Fig. 23, we present results for density and interface reconstruction in the same way as in Fig. 22 for the ST multimaterial ALE code. Conclusions from this figure are the same as for the ST code: Results for both remapping methods are identical to the naked eye.

The CC ALE code takes $9.65 \cdot 10^3\, s$ to run using hybrid remapping and $3.51 \cdot 10^4\, s$ to run using intersection-based remapping for a resolution comparable with 17×100 resolution on a rectangular grid, so the intersection-based method is about 3.63 times more expensive than the hybrid method. For the ST ALE code and similar spatial resolution the intersection-based method was approximately 1.2 more expensive than the hybrid method.

We conclude this section by presenting pictures of interface reconstruction and mixed points as defined in the description of hybrid remapping for initial and final time, see Fig. 24. We observe that during the calculation the number of mixed points is changing because the “length” and complexity of the interface are growing. This is one of the reasons why it is practically impossible to have an estimate of CPU time per cell for a realistic ALE calculation.

6.4. Shock-Bubble Interaction

This test case is taken from the paper [29] (see also, [9]). The setup consists of a Helium bubble surrounded by air. The initial domain is the rectangular box $[0, 0.650] \times [-0.089, 0.089]$,

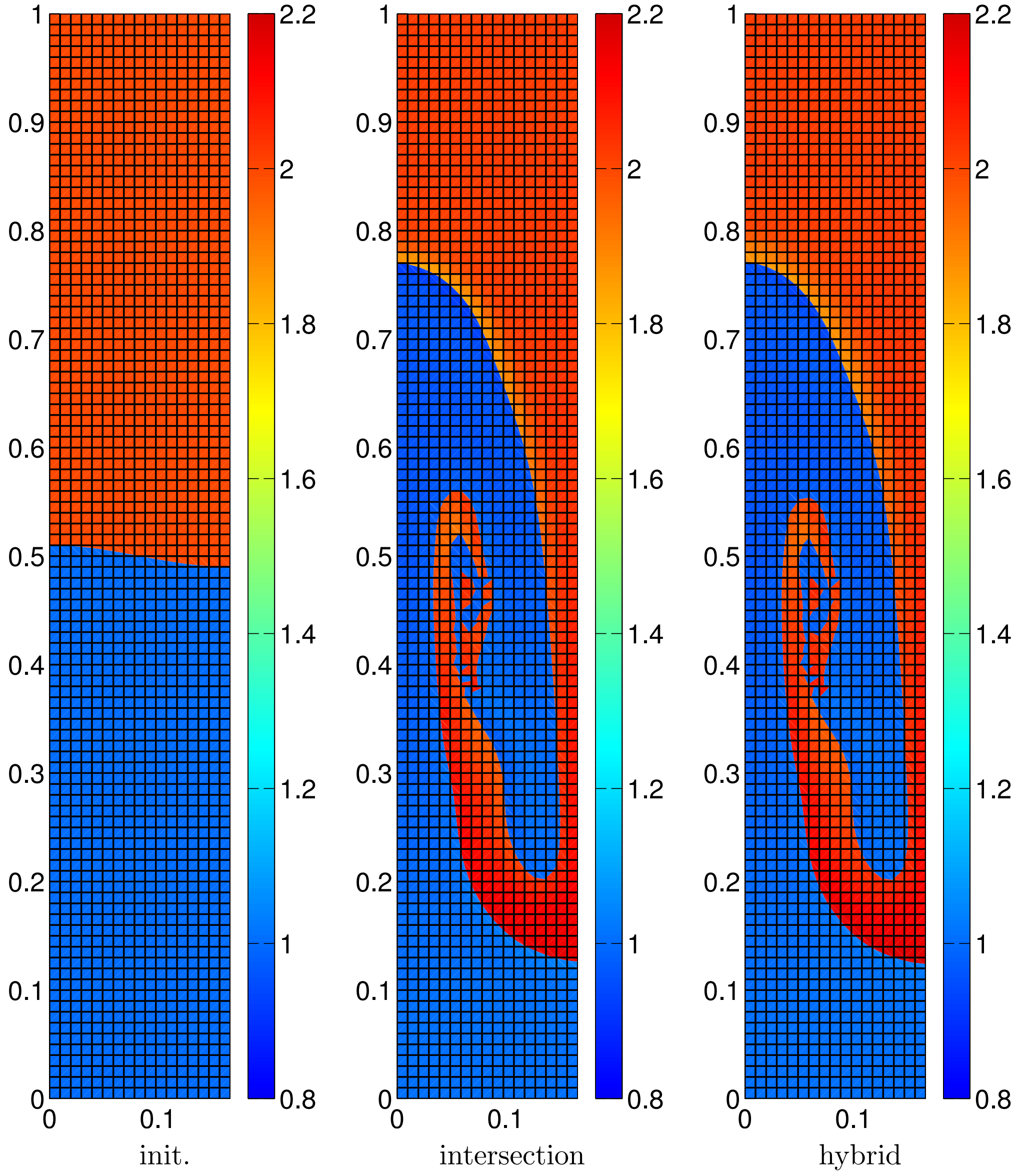


Figure 22: "Multimaterial" Rayleigh-Taylor instability problem - results for the ST multimaterial ALE code. Two left panels represent initial density; central panel represents density for intersection-based remapping at the final time; right panel represents density for hybrid remapping at final time.

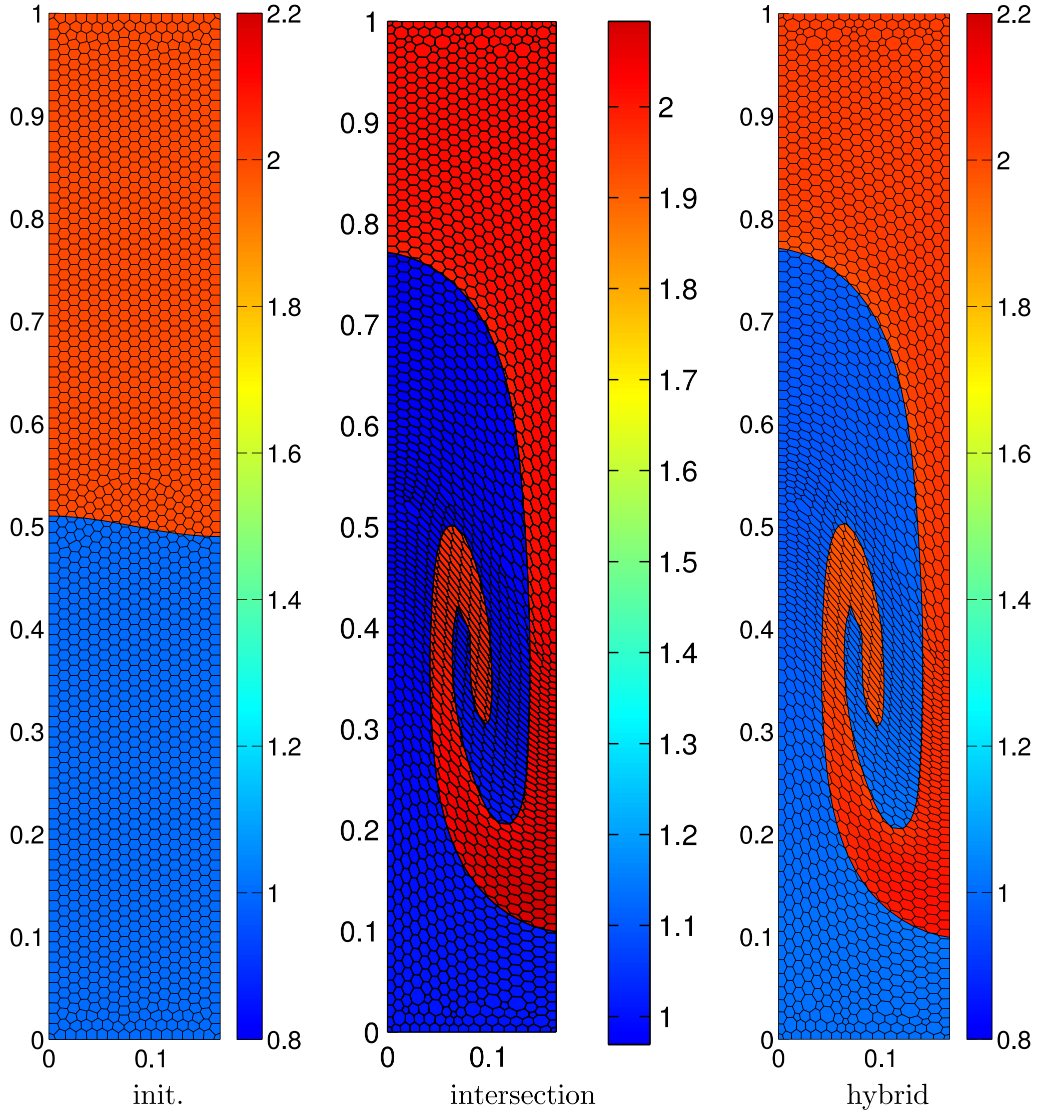
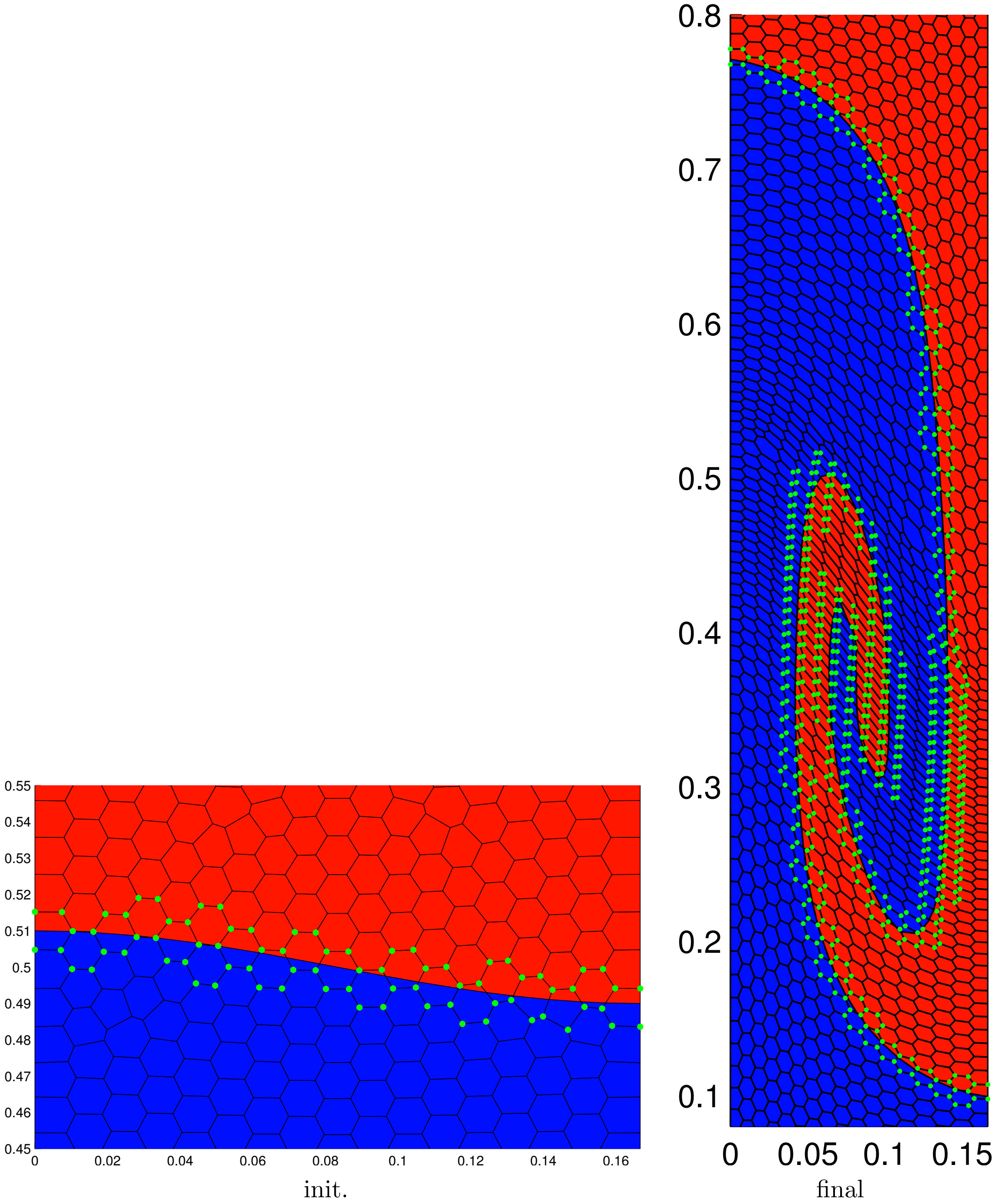


Figure 23: “Multimaterial” Rayleigh-Taylor instability problem - results for the CC multimaterial ALE code. Left panel represents initial density; central panel represents density for intersection-based remapping at the final time; right panel represents density for hybrid remapping at final time.



init.

final

Figure 24: Fragments of initial and final grid. Points marked as "mixed", at initial and final moment - CC code.

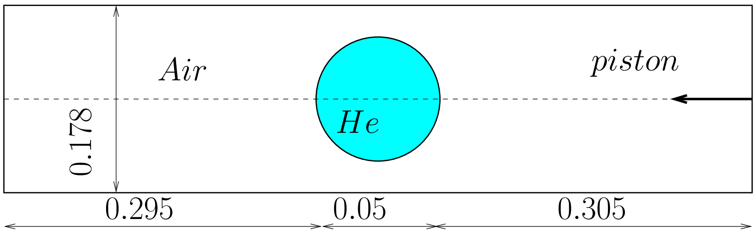


Figure 25: Computational domain for shock-bubble interaction problem.

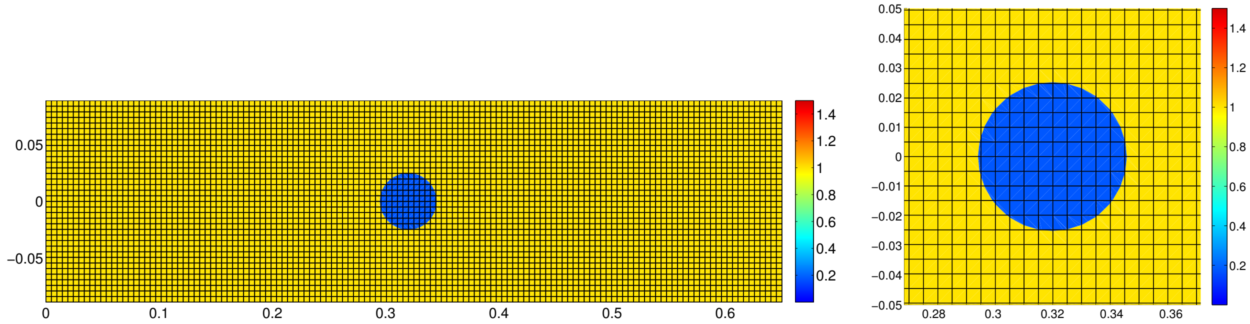


Figure 26: Initial density distribution for shock-bubble interaction problem - the entire domain and zoom of the region around bubble - ST code.

Fig. 25. The bubble is a circle defined by the coordinates of its center $(x_c, y_c) = (0.320, 0)$ and its radius $R_b = 0.025$. We prescribe wall boundary conditions on all boundaries with the exception of the right boundary, where we impose a piston-like boundary condition defined by the inward velocity $\boldsymbol{V}^\star = (u^\star, 0)$. The incident shock wave produced by motion of the piston is defined by its Mach number, $M_s = 1.22$. The bubble and the air are initially at rest. The initial data for Helium are $(\rho_1, P_1) = (0.182, 10^5)$, its molar mass is $\mathcal{M}_1 = 5.269 \cdot 10^{-3}$ and its adiabatic constant is $\gamma_1 = 1.648$. The initial data for air are $(\rho_2, P_2) = (1, 10^5)$, its molar mass is $\mathcal{M}_2 = 28.963 \cdot 10^{-3}$ and its adiabatic constant is $\gamma_2 = 1.4$. Using the Rankine-Hugoniot relations, we find that the x-velocity of the piston is given by $u^\star = -124.824$. The x-component of the incident shock velocity is $D_c = -456.482$. The incident shock wave hits the bubble at time $t_i = 668.153 \cdot 10^{-6}$. The stopping time for our computation is $t_{end} = t_i + 674 \cdot 10^{-6} = 1342.153 \cdot 10^{-6}$. It corresponds to the time for which an experimental shadow-graph is displayed in [29].

This problem is a two-material problem and to model it, we need to use a multimaterial ALE code. As for the previous examples, we will present results both for the ST multimaterial ALE code as well as for the CC multimaterial ALE code. Because this is a true multimaterial problem, we compare only intersection-based and hybrid remapping. Similar to the previous example, we do not present results for interface reconstruction because they are nearly identical to the color map of density.

We now present results obtained by the ST code. We chose to use an initial rectangular grid with $134 \times 36 = 4824$ cells, and to run this problem in real ALE mode smoothing the grid and remapping after every Lagrangian step. Initial density distribution at the initial time is depicted in Fig. 26.

Density distributions for the final time using intersection-based and hybrid remapping are depicted in Fig. 27. There is a slight difference in the results at x axes.

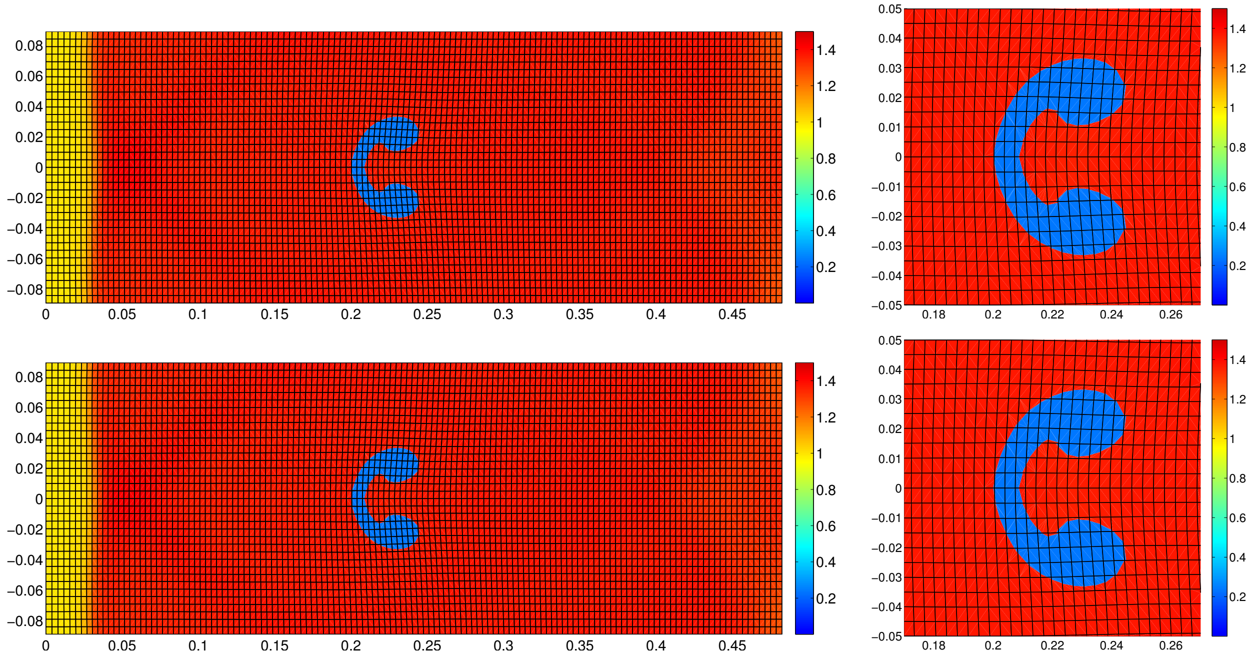


Figure 27: Density distribution for shock-bubble interaction problem - final time - the entire domain and zoom of the region around bubble - ST code. Upper row – intersection-based remap, lower row – hybrid remapping.

These figures demonstrate that results obtained by intersection-based and hybrid remapping are practically the same.

It takes about 2078 s to run this calculation using the intersection based method and about 1529 s to run it using hybrid remapping. Thus, the intersection-based method is about 35% more expensive than hybrid method, which is a similar result to the multimaterial Rayleigh-Taylor instability problem.

Now, we present results obtained by using the CC multimaterial ALE code. We chose to run this code in real ALE mode using grid smoothing and remapping after each Lagrangian step. The initial grid is polygonal consisting of 4847 polygons and is depicted in Fig. 28.

The initial density distribution at the initial time is depicted in Fig. 28.

Density distributions at the final time that were obtained using intersection-based as well as hybrid remapping are presented in Fig. 29. There is a slight difference in the results at x axes.

These figures demonstrate that results obtained by intersection-based and hybrid remapping are practically the same.

The CC multimaterial ALE code takes $3.51 \cdot 10^4$ s to run this problem using intersection-based remap and $9.65 \cdot 10^3$ s using hybrid remapping. So, intersection-based remapping is about 3.64 times more expensive than hybrid remapping. This result is similar to the multimaterial Rayleigh-Taylor instability problem.

7. Conclusions and Discussion

In this paper, we have presented a new hybrid rezoning/remapping algorithm for multimaterial ALE methods. Hybrid remapping is performed in two steps. In the first step only nodes of the grid are moved that are inside subdomains occupied by single materials. At this stage, computationally

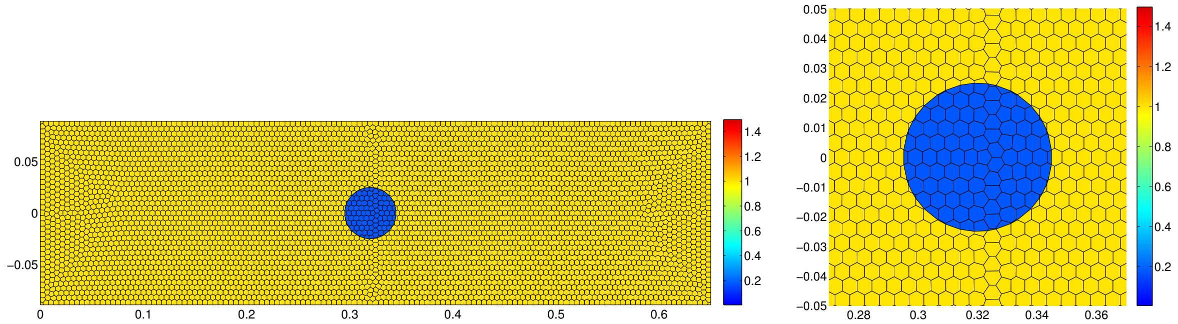


Figure 28: Initial density distribution for the shock-bubble interaction problem - the entire domain and zoom of the region around bubble - CC code.

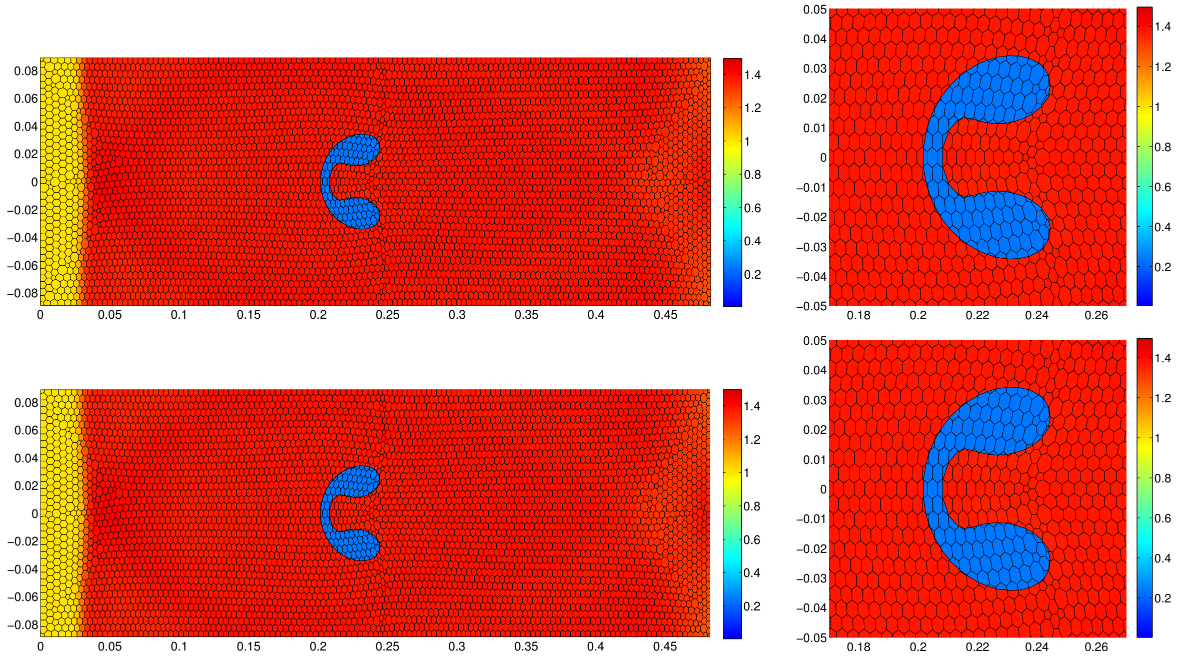


Figure 29: Density distribution for the shock-bubble interaction problem - final time - the entire domain and zoom of the region around bubble - CC code. Upper row – intersection-based remap, lower row – hybrid remapping.

cheap swept-region remapping is used. In the second step, nodes are moved that are vertices of mixed cells or vertices of some cells in the buffer zone around mixed cells. At this stage, intersection-based remapping is used. We have demonstrated the performance of this new method for both structured and unstructured polygonal grids in two dimensions as well as for ALE codes using staggered and cell-centered discretizations.

We have shown that the accuracy of intersection-based remapping and hybrid remapping is almost the same. This is demonstrated on a cyclic stand-alone remapping example, where the exact solution is known, as well as on calculations of two realistic problems - a Rayleigh-Taylor instability and an interaction of a shock with a gas bubble.

We showed that CPU time strongly depends on implementation of the intersection algorithm and other implementation issues.

We now discuss some considerations which must be taken into account when considering the efficiency of remapping in multimaterial ALE codes.

First, it is necessary to estimate the cost of the hydro part relative to other physics that is implemented in the code. In the case of multiphysics codes, which include hydrodynamics, strength of materials, radiation hydrodynamics, astrophysics, combustion and so on, [28], [31], [26], [5], [35], the relative computational cost of the hydro part may be small. In this situation, accuracy is the most important issue and, thus, intersection-based remap should be used everywhere. In multiphysics codes many quantities must be remapped. In this situation the relative cost of intersection based remap is lower because intersections are done only once and used for many quantities - the cost of remap is dominated by reconstruction of all variables and by computing fluxes, which has to be done for swept-region remapping, too. LASNEX, [35], is an example of a multiphysics code where intersection-based remapping is used for the entire grid.

Another consideration is the cost of remapping in ALE itself relative to the cost of the Lagrangian and rezoning stages. This depends on the type of grid the code is using: structured, block structured, or unstructured, and on the type of data structures that are used to represent it. One also needs to take into account the type of discretization the code is using: staggered or cell-centered. Another consideration is how many materials are used in the problem, which affects the fraction of the computational domain that is occupied by mixed cells. All these considerations may affect performance of the algorithm and in particular cost of memory access.

For 2D codes it is important to know if the Lagrangian step allows grid movement that creates valid non-convex cells or not. If so, then for intersection based remap more complicated and expensive intersection algorithms must be used, for example, based on subdivision of the cell into triangles, [9]. In 3D, subdivision of cells into tetrahedra appears to be unavoidable. Therefore, even for codes with relatively simple physics, the choice of remapping algorithm can be different in 2D and 3D.

Finally, the computer architecture on which algorithm will be implemented and available parallelization strategies must be considered (see, [34, 33], for examples of implementations of compressible gas dynamics on the Roadrunner machine).

Considerations presented in this section, the analysis of relative cost of multimaterial remapping performed in section 5.4, and timing results presented in section 6 for two different multimaterial ALE codes suggest that in multiphysics codes all options should be available, such that different remapping methods can be used for different problems and/or on different computer platforms.

Acknowledgment

This work was performed under the auspices of the National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396 and supported by the DOE Advanced Simulation and Computing (ASC) program. The authors acknowledge the partial support of the DOE Office of Science ASCR Program. Milan Kucharik was supported by the Czech Ministry of Education grants MSM 6840770022, MSM 6840770010 and LC528, and the Czech Science Foundation project P201/10/P086. Partial support of the BARRANDE MEB021020 program is also gratefully acknowledged. The authors thank A. Barlow and R. Loubère for stimulating discussions over many years.

References

[1] H. T. Ahn and M. Shashkov. Multi-material interface reconstruction on generalized polyhedral meshes. *J. Comput. Phys.*, 226:2096–2132, 2007.

[2] D. J. Benson. An efficient, accurate, simple ALE method for nonlinear finite element programs. *Comp. Meth. Appl. Mech. Engrg.*, 72:305–350, 1989.

[3] D. J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Comp. Meth. Appl. Mech. Engrg.*, 99:235–394, 1992.

[4] D. J. Benson. Momentum advection on a staggered mesh. *J. Comput. Phys.*, 100:143–162, 1992.

[5] J. I. Castor. *Radiation Hyddrodynamics.* Cambridge University Press, 2004.

[6] Q. Du, V. Faber, and M. Gunzburger. Centroidal Voronoi tesselations: applications and algorithms. *SIAM Review*, 41:637–676, 1999.

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

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

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

[10] C. W. Hirt, A. Amsden, and J. L. Cook. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *J. Comput. Phys.*, 14:227–253, 1974.

[11] P. Hoch. An arbitrary Lagrangian-Eulerian strategy to solve compressible fluid flows. HAL : hal-00366858, version 1, 2009. Available at http://hal.archives-ouvertes.fr/docs/00/36/68/58/PDF/ale2d.pdf.

[12] P. Hoch. Mesh quality and conservative projection in Lagrangian compressible hydrodynamic. Conference on Numerical methods for multi-material fluid flows; Czech Technical University in Prague on September 10 - 14, 2007; http://www-troja.fjfi.cvut.cz/∼multimat07/presentations/tuesday/Rebourcet_Hoch.pdf.

[13] D. S. Kershaw, M. K. Prasad, M. J. Shaw, and J. L. Milovich. 3D unstructured mesh ALE hydrodynamics with the upwind discontinuous finite element method. *Comp. Meth. Appl. Mech. Engrg.*, 158:81–116, 1998.

[14] P. Kjellgren and J. Hyvarien. An arbitrary Lagrangian-Eulerian finite element method. *Computational Mechanics*, 21:81–90, 1998.

[15] P. Knupp, L. G. Margolin, and M. Shashkov. Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods. *J. Comput. Phys.*, 176:93–128, 2002.

[16] M. Kucharik, J. Breil, S. Galera, P.-H. Maire, M. Berndt, and M. Shashkov. Hybrid remap for multi-material ALE. *Computers & Fluids*, 2010. Accepted for publication.

[17] M. Kucharik, R. Garimella, S. Schofield, and M. Shashkov. A comparative study of interface reconstruction methods for multi-material ALE simulations. *J. Comput. Phys.*, 229:2432–2452, 2010.

[18] M. Kucharik and M. Shashkov. Multi-material remap for staggered ALE in 2D. Conference on Numerical methods for multi-material fluids and Structures; Pavia, Italy, September 21 - 25, 2009; `http://www.eucentre.it/index.php/content/view/1287/415/lang,it/kucharik.pdf`.

[19] M. Kucharik and M. Shashkov. Conservative multi-material remap for staggered discretization. 2010. In preparation.

[20] M. Kucharik, M. Shashkov, R. Liska, P. Maire, M. Berndt, and M. Shashkov. Hybrid remap for multi-material ALE, 2010. Accepted for publication.

[21] M. Kucharik, M. Shashkov, and B. Wendroff. An efficient linearity-and-bound-preserving remapping methods. *J. Comput. Phys.*, 188:462–471, 2003.

[22] R. Loubère and M. Shashkov. A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian-Eulerian methods. *J. Comput. Phys.*, 204(23):155–160, 2004.

[23] H. U. Mair. Review: Hydrocodes for structural response to underwater explosions. *Shock and Vibrations*, 6:81–96, 1999.

[24] L. G. Margolin. Introduction to "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds". *J. Comput. Phys.*, 135:198–202, 1997.

[25] L. G. Margolin and M. Shashkov. Second-order sign-preserving conservative interpolation (remapping) on general grids. *J. Comput. Phys.*, 184(1):266–298, 2003.

[26] J. W. Murphy and A. Burrows. BETHE-HYDRO: An arbitrary Lagrangian-Eulerian multidimensional hydrodynamics code for astrophysical simulations. *Astrophysical Journal Supplement Series*, 179:209–241, 2008.

[27] J. S. Peery and D. E. Carroll. Multi-material ALE methods in unstructured grids. *Comp. Meth. Appl. Mech. Engrg.*, 187:591–619, 2000.

[28] D. Post. Codes written by the national and international computational physics community. Technical Report LA-UR-02-6284, Los Alamos National Laboratory, 2002. Available at `http:// www.highproductivity.org/026284coverCEEGodes.pdf` .

[29] J. Quirk and S. Karni. On the dynamics of a shock-bubble interaction. *J. Fluid Mech.*, 318:129–163, 1996.

[30] M. Shashkov. Closure models for multimaterial cells in arbitrary Lagrangian-Eulerian hydrocodes. *Int. J. Numer. Meth. Fluid*, 56:1497–1504, 2007.

[31] D. J. Torres and M. F. Trujillo. KIVA-4: An unstructured ale code for compressible gas flow with sprays,. *J. Comput. Phys.*, 219:943–975, 2006.

[32] A. Winslow. Numerical solution of the quasilinear Poisson equations in a nonuniform triangle mesh. *J. Comput. Phys.*, 1:149–172, 1966.

[33] J. G Wohlbier, R. B. Lowrie, and B. Bergen. Acceleration of a multi-material hydrodynamics algorithm on roadrunner. Conference on Numerical methods for multi-material fluids and Structures; Pavia, Italy, September 21 - 25, 2009; `http://www.eucentre.it/index.php/content/view/1287/415/lang,it/wohlbier.pdf`.

[34] P. R. Woodward, J. Jayaraj, P.-H. Lin, and W. Dai. First experience of compressible gas dynamics simulation on the Los Alamos Roadreanner machine. *Concurency and Computation: Practice and Experience*, 21:2160–2175, 2009.

[35] G. Zimmerman, D. Kershaw, D. Bailey, and J. Harte. LASNEX code for inertial confinement fusion. *J. Opt. Soc. Am.*, 68:549, 1978.