

LA-UR-07-0656

Approved for public release;  
distribution is unlimited.

*Title:* Multi-material interface reconstruction on  
generalized polyhedral meshes

*Author(s):* Hyung Taek Ahn (htahn@lanl.gov)  
Mikhail Shashkov (shashkov@lanl.gov)  
Theoretical Division, Group T-7  
Los Alamos National Laboratory  
Los Alamos, NM 87545, U.S.A.

*Intended for:* Submission to the Journal of Computational Physics



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

# Multi-material interface reconstruction on generalized polyhedral meshes

Hyung Taek Ahn and Mikhail Shashkov

January 23, 2007

## Abstract

We describe multi-material (more than two materials) interface reconstruction methods for 3D mesh of generalized polyhedrons. Basic information used in interface reconstruction is volume fractions of each material in mixed cell containing several materials. All methods subdivide mixed cell into set of pure non-overlapping sub-cells each containing just one material that have reference volume fraction. We describe three methods. First two methods represent extension of standard piece-wise linear interface construction (PLIC) methods into 3D and use information only about volume fractions. First method is first-order accurate and based on discrete gradient of volume fraction as estimate to normal to interface. Second method is planarity preserving (second-order accurate) and is extension to 3D of least squares volume of fluid interface reconstruction algorithm (LVIRA, see [46, 45] for 2D case). The third method is extension to 3D of so-called moment of fluid (MoF) method, [14, 15]. MoF method is second-order accurate. This method uses information not only about volume fraction but also about position of the centroids for each material. In contrast to standard PLIC methods, MoF method uses only information from the cell where reconstruction is performed, no information from neighboring cells is needed. Also MoF method provides automatic ordering of the materials in the process of interface reconstruction. Optimal ordering is based on comparing positions of the reference and actual (centroids of reconstructed pure sub-cells) positions of the centroids. Performance of the methods is demonstrated on numerical examples. In appendices we describe in detail all geometrical and optimization algorithms needed for implementation of interface reconstruction methods.

# 1 Introduction and Background

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 [25], 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 Lagrangian and Eulerian methods. This class of methods has been termed Arbitrary Lagrangian-Eulerian or ALE. Many authors have described ALE strategies to optimize accuracy, robustness, or computational efficiency, see for example [3, 35, 28, 44, 33].

It is possible to formulate the ALE scheme as a single algorithm [13] based on solving the equations in a moving coordinate frame. For multi-material flows it is usual to separate ALE scheme into three separate stages. These are: 1) a Lagrangian stage in which the solution and 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.

We are interested in developing an ALE methodology for 3D unstructured mesh consisting of *generalized* polyhedra for high-speed multi-material flows with strong shear deformations, which occur in many problems of interest. In this paper, a generalized polyhedron is a 3D solid with arbitrary topology and, possibly, non-planar polygonal faces. This general consideration is necessary because even simple polyhedra such as hexahedra can start out with planar faces at the start of the simulation but end up with non-planar faces due to the movement of its nodes as induced by the flow. Clearly, the geometry of a non-planar polygon is not uniquely defined. We will consider the issues raised by the presence of such faces in Section 2. The use of an unstructured mesh consisting of generalized polyhedra simplifies the setup process for computational domains with complex geometrical shapes and helps to avoid artificial mesh imprinting due to the restrictions of a conventional mesh consisting only of tetrahedra and generalized bricks, [9, 8].

For multi-material flows initial mesh is usually aligned with material interface, that is, each cell of the mesh contains only one material. For simple flows, it is possible to rezone the mesh in each material and keep interface aligned with the mesh, that is, do not move nodes on interface at all or move it along the interface. Due to the nature of shock wave propagation in complex materials for high-speed multi-material flows with

strong shear deformations, the ALE methods are currently the only proven technology to solve such problems. In ALE methods, the mesh does not move with the fluid, and so it is unavoidable that mixed cells containing two or more materials will appear.

Multi-material cells in ALE methods represent material interfaces that undergo high deformation. The main difficulties in this case are 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 multi-material cells. A separate set of material properties is normally maintained for all the materials in each multi-material cell along with the volume fractions that define the fraction of the cell's volume occupied by each material.

A sub-cell model is then required to define how the volume fractions and states of the individual materials evolve during the Lagrangian step, [2, 6, 59, 11, 12, 54, 36, 41]. This sub-cell model is required to close the governing equations, which otherwise are underdetermined. Most of the sub-cell closure models do not require positions of the interfaces inside the mixed cells, however more modern methods like one presented in [2, 11] do require interface locations and orientations, and therefore require some representation of the interface.

In next, rezoning stage, the nodes of the computational grid are moved to a more optimal position. For purpose of this paper it is not important what algorithm is used. Interested reader can find some review of rezoning algorithms in [29]. Rezoning stage results in the new mesh. To start new Lagrangian step we need to conservatively interpolate all flow parameters from Lagrangian mesh on completed time step to new rezoned mesh. This process is called remapping.

Remapping for case when there is only one material (and therefore there are no mixed cells) is described, for example, in [34, 32, 18]. In situation when several materials present, even if after Lagrangian step all cells were pure (containing only one material), rezone mesh will have mixed cells. Therefore, in remap stage one need to determine which cells of rezone mesh are pure and which are mixed, and in particular, find parameters for each material in mixed cells. There are several approaches to perform multi-material remap, but all of them require knowledge of the interface on Lagrangian mesh from completed time step. Therefore, interface has to be represented in some way on the mesh after Lagrangian step.

There are several well established methods for dealing with interfaces: volume of fluid (VoF) method (which uses interface reconstruction), [26, 48, 5]; front tracking, [61, 22, 60]; level set,[56, 55, 43].

For modeling three dimensional high-speed compressible multi-material (more than two materials) flows, with topology of interface changing in time, and when exact conservation is critical, on general meshes, VoF seems to be method of choice, [5, 4, 3, 52].

Originally VoF was developed for modeling of the dynamics of incompressible flows with free boundaries using Eulerian approach, [26]. The basic object of VoF methods is a *two material medium*. Typical VoF method consists of two steps: interface reconstruction (using volume fractions) and update of the volume fractions in time. Excellent reviews of VoF methods and, in particular, interface reconstruction methods can be found in the following papers [48, 5, 45, 50, 49]. In this paper we are only interested in interface reconstruction.

The most common interface representation used by interface reconstruction methods consists of a single linear interface (line in 2D and plane in 3D) per mixed cell containing two materials. This class of interface representation commonly called Piecewise-Linear Interface Calculation (PLIC). The location of linear interface, for a given volume fraction, uniquely defined by direction of the interface outward normal. There are a number of ways to define the direction of the normal, [14]. We will describe some of them in Section 3.1.

If mixed cell is convex polygon in 2D, or convex polyhedron in 3D then each material in mixed cell will be represented by convex sub-polygon or convex sub-polyhedron (for simplicity, we will call both of the by sub-cells), which is obtained by intersection of corresponding half-plane or half-space with mixed cell. If mixed cell is non-convex, than material in mixed cell can be represented by disjoint pieces. 2D examples of mixed cells with linear interface are delineated in Fig. 1 both for convex and non-convex cases.

The standard interface reconstruction methods for two materials use only information about volume fractions. Recently, we have developed new method, *moment of fluid* or MoF method, [14, 15], which also uses information about position of the centroid of the material in the mixed cell. In MoF method objective is to minimize distance between specified and actual centroids obtained as result of reconstruction, when exactly matching specified volume fraction. In 2D it leads to minimization of one dimensional function, [14].

Here we want to mention that there are hydrocodes, where centroids of the materials are tracked, see for example, [23, 1] for LASNEX code description, as well as description of some algorithms for transporting centroids in finite element hydrocodes in [5, 4].

In case of more than two materials we are only interested in algorithms,

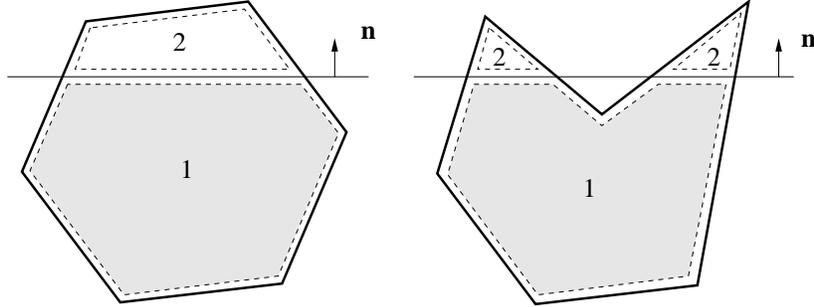


Figure 1: Two material interface reconstruction in 2D. Left shows convex polygonal cell, and right corresponds to non-convex case. If the original cell is non-convex, the sub-cells after intersection remain non-convex.

where each material will be represented by collection of pure sub-cells (possibly disjoint). These sub-cells do not intersect with each other, and union of all sub-cells representing all materials is equal to entire mixed cell. The total volume of all sub-cells representing material of one kind has to match corresponding volume fraction. It is interesting to know that in most VoF algorithms interface reconstruction procedure does not satisfy this property, [5, 58], because they use so-called *onion-skin* model, [64], [5], which may lead to intersection of sub-cells representing different materials. The problems related to this phenomenon are usually resolved on advection stage of VoF methods by limiting fluxes, e.g., [58, 16]. The modern methods produce sub-cells representing different materials, which do not intersect each other, [39, 37, 15].

In standard interface reconstruction methods multi-material case is dealt with by sequentially applying algorithm for two materials, which assumes some externally specified material ordering, which we will discuss later.

To the best of author's knowledge there are only two approaches where for multi-material interface reconstruction does not require ordering. First method introduced in [7] uses innovative extension of contouring algorithm to the case of many materials. This method does not reproduce specified volume fractions exactly. Second method is based on use of particles and power diagrams, [51], is very promising but requires more work to be considered for practical applications. We do not consider these methods in our paper.

Following [15] we will call algorithm, which is based on sequential application of algorithm for two materials and which satisfies our requirements

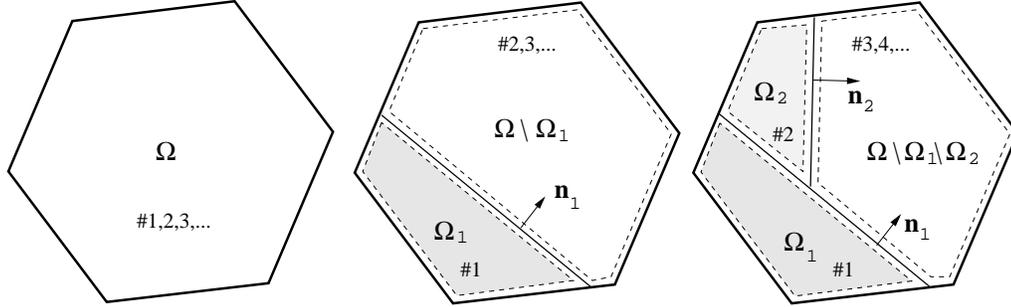


Figure 2: Nested dissection strategy on 2D polygonal cell. Three consecutive stages of nested dissection: left—three material mixed cell, middle—after first dissection, right—after second dissection.

by *nested dissection*.

General idea that materials are cut from mixed cell one by one in specified order as illustrated in Fig. 2 at each step of the sequence. Let us denote mixed cell by  $\Omega$ . First material #1 is separated from the rest of the materials in mixed cell. It can be done by any interface reconstruction methods for two material case, where first material is material #1 and second material is union of all other materials. This first step creates pure sub-cell (or several pure sub-cells in case on non-convex mixed cell) which represent material #1. We will denote this set by  $\Omega_1$ . After this step material #1 is removed from mixed cell and we consider set  $\Omega \setminus \Omega_1$ . This set contains materials #2, 3, ... and so on. Next step is to remove material #2 from set  $\Omega \setminus \Omega_1$ . This again can be done by applying two material algorithm to  $\Omega \setminus \Omega_1$ , where first material will be material #2 and the second material will be union of materials #2, 3, ... and so on. One need to understand that volume fraction of material #2 has to be recomputed with respect to set  $\Omega \setminus \Omega_1$ . This second step creates pure sub-cells representing material #2,  $\Omega_2$ . After this step material #2 is removed from mixed cell and we consider set  $\Omega \setminus \Omega_1 \setminus \Omega_2$ . This process is repeated till we create pure sub-cells for all materials.

There were several attempts to develop automatic local schemes for material ordering [47, 39, 4, 1] for standard PLIC methods. All of them are based on using information about exact or approximate of centroid position of the materials in mixed cell itself and its neighbors. To best of our knowledge, e.g, examples in [15], none of these methods treat even all basic cases correctly.

In MoF method positions of centroids for materials inside mixed cell is

used not only for each two material interface reconstruction step, but also for choosing the best material ordering that minimizes discrepancy between given and actual centroids computed from reconstructed pure sub-cells for all materials, [15].

Here we also mention recent paper [10] that deals with very special, but important case of triple point configuration in 2D on square mesh. Algorithm presented in [10] is very specialized and assumes that first material in reconstruction process is externally specified; position of second interface is obtained from optimization procedure.

Most of the papers related to interface reconstruction are dealing with two dimensional case, there are only several papers which describe interface reconstruction in 3D, almost all of them dealing with brick mesh and case of two materials, e.g., [31], and references herein. In [31] 3D extension of so-called *Swartz-Mosso*, [57, 40, 14], algorithm is developed, but it is described only for orthogonal mesh, however authors claim that it can be easily extended to unstructured and distorted meshes. Description of 3D interface reconstruction on distorted and unstructured meshes are very rare and it is usually unpublished reports related to ALE methods, papers published in proceedings of conferences, e.g., [30], or buried in some papers related to particular application, e.g., [17]. We refer interested reader to [16] for some details related to interface reconstruction for two material case on general hexahedral meshes and for further references.

The goal of this paper is to extend MoF method for multi-material interface reconstruction to the case of 3D mesh consisting of generalized polyhedrons. We also present possible extensions of two standard methods for interface reconstruction for two materials: method based on estimation of gradient of volume fraction, and least squares volume-of-fluid interface reconstruction algorithm (LVIRA), [46, 45] to the case of multi-material interface reconstruction on 3D mesh consisting of generalized polyhedrons. We present all details of algorithms, which will allow interested reader to reproduce them.

The outline of the rest of this paper is as follows. In Section 2 we describe class of 3D meshes consisting of generalized polyhedrons and corresponding data structures. In next Section 2.2 we describe how materials are represented in multi-material mixed cell. In Section 3 we give general description of three interface reconstruction methods for case of two materials: volume fraction gradient based (GRAD) method, LVIRA, and MoF, and present numerical examples for two material interface reconstruction. Geometric algorithms needed for interface reconstruction are described in Appendix A. Optimization algorithms needed for implementation of LVIRA and MoF

methods are described in Appendix C. Initialization of volume fractions and centroids is described in Appendix D. In Section 4, we describe extensions of GRAD, LVIRA and MoF methods to multi-material case and present examples of multi-material interface reconstruction. Finally, we conclude in Section 5.

## 2 Multi-material interface representation in generalized polyhedral mesh

### 2.1 Definition of the generalized polyhedral mesh (GPM)

In this paper, we are interested in the mesh which consist of *generalized polyhedrons - generalized polyhedral mesh - (GPM)*. The generalized polyhedron can be thought as 3D solid obtained from polyhedron by perturbing positions of its vertices, which makes faces non-planar. As we have mentioned in Introduction generalized polyhedrons naturally appear in ALE calculations, because even if initial mesh consist of classical polyhedra such as hexahedra with planar faces at the start of the simulation after Lagrangian or rezone step of ALE faces maybe non-planar due to the movement of its vertices.

The geometry of a face whose vertices are not all in a single plane, however, is not unique. Therefore, we adopt a faceted representation to obtain a consistent definition of its geometry as follows (see,[18], for more detail). First, we define the “face center”

$$\mathbf{x}_c^{face} = \frac{\sum_{\mathbf{x}_i \in \mathcal{P}(f)} \mathbf{x}_i}{|\mathcal{P}(f)|} \quad (1)$$

where  $\mathcal{P}(f)$  is the set of all face vertices,  $|\mathcal{P}(f)|$  is number of vertices for the face, and  $\mathbf{x}_c^{face}$  and  $\mathbf{x}_i$  are the coordinates of face center and  $i$ -th vertex respectively. Next, the faces of the generalized polyhedral cell is triangulated by using the face center and two vertices of each edge. In the second step, the triangulated generalized polyhedral cell is decomposed into sub-tetrahedra by using triangulated surface and one additional vertex inside of the cell, called “cell center” defined as follows

$$\mathbf{x}_c^{cell} = \frac{\sum_{\mathbf{x}_i \in \mathcal{P}(c)} \mathbf{x}_i}{|\mathcal{P}(c)|} \quad (2)$$

where  $\mathcal{P}(c)$  is the set of all cell vertices,  $|\mathcal{P}(c)|$  is total number of vertices for the cell, and  $\mathbf{x}_c^{cell}$  and  $\mathbf{x}_i$  are the coordinates of cell center and  $i$ -th vertex respectively. Hence,  $m$ -verticed polygonal face is divided into  $m$ -triangles,

and  $n$ -faced polyhedron is further decomposed into  $n \times m$  sub-tetrahedra provided that each face is  $m$ -verticed. For example, the generalized polyhedral representation of a hexahedral cell results in  $6 \times 4$  sub-tetrahedra as displayed in Fig. 3.

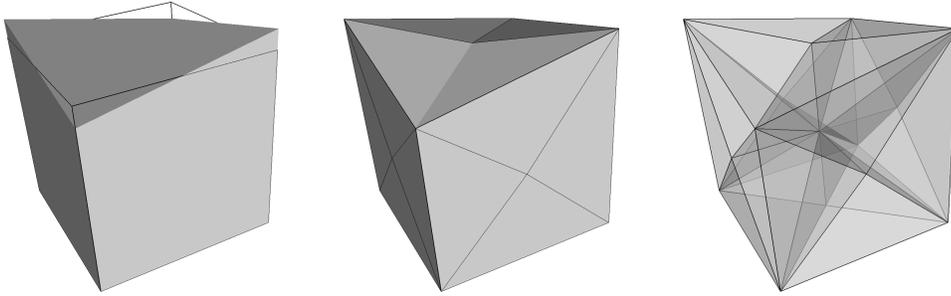


Figure 3: Generalized polyhedral representation of hexahedron with non-planar face. Left figure shows initial hexahedral cell with non-planar top face (wire frame of a cube is overlapped to emphasize the non-planar top face), middle shows surface triangulation of the hexahedron, and right shows sub-cell decomposition of the hexahedron.

Representation of computational cell as collection of the tetrahedra simplifies implementation of algorithms of computational geometry. In 2D analog of this process is to represent cell as collection of the triangles Fig. 4. It is especially useful for non-convex cells, which can easily appear in ALE calculations.

This generalization of polyhedral cell brings two merits. First, the planar face restriction of polyhedral cell is relaxed so that it can has vertices not always on a plane. Second, it allows us to deal with non-convex cells (which are always present in 3D), as long as the cell can be decomposed into valid sub-cells (cell center,  $\mathbf{x}_c^{cell}$ , can “see” all the vertices). These features are advantageous for dealing with meshes in ALE methods.

## 2.2 Interface representation in generalized polyhedral cells

First, we illustrate the representation of the multi-material mixed cell in 2D. As we have discussed in previous section each cell of the mesh in 2D

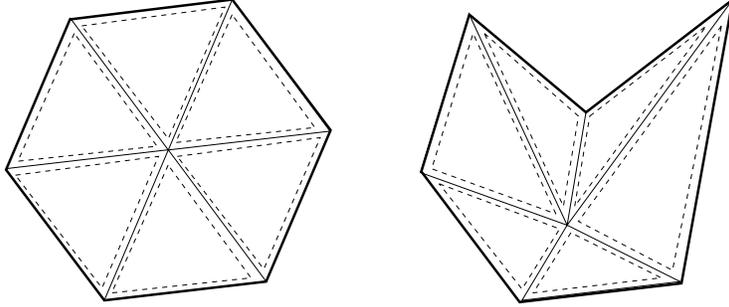


Figure 4: Representation of the polygonal cell in 2D. Sub-cell (indicated by dashed lines) structure of *pure* generalized polygonal cell is illustrated for convex (left) and non-convex (right) case. Notice that sub-cells are always convex regardless of convexity of the original cell.

is decomposed into convex sub-cells (triangles) as shown in Fig. 4. General scheme for multi-material interface reconstruction is described in Introduction and consist in nested dissection of the original cell. First step of this process is to define part of the cell which is occupied by material #1. Cell is intersected with half-plane (specified by normal  $\mathbf{n}$ ), such that this intersection has prescribed volume (volume fraction), Fig. 5. In Fig. 5 part of the cell occupied by material #1 is shaded. Technically material #1 is represented as collection of shaded polygons, each of which is part of one of the triangles of original triangulation of the cell. These polygons are intersections of half-plane with corresponding triangles. Sum of the volumes of these polygons matches prescribed volume (volume fraction) for material #1. Here we need to mention that for each cell we have to match only *one* volume fraction. There are no independent volume fractions are specified for each triangle.

Materials #2,#3 and so on are contained in non shaded polygons, which are also parts triangles of original triangulation. Following to nested dissection scheme process continues to dissect of non shaded polygons to find part of the cell which represents material #2. It easy to see that material #2 will be represented by collection of the convex polygons which are intersections of corresponding half-plane with non shaded polygons obtained on first step. To find convex polygons representing material #3 one need to repeat the process.

In 3D process is the same, one just need to substitute triangles by tetrahedra, and half-plane by half-space, Fig. 6. Three different types of the

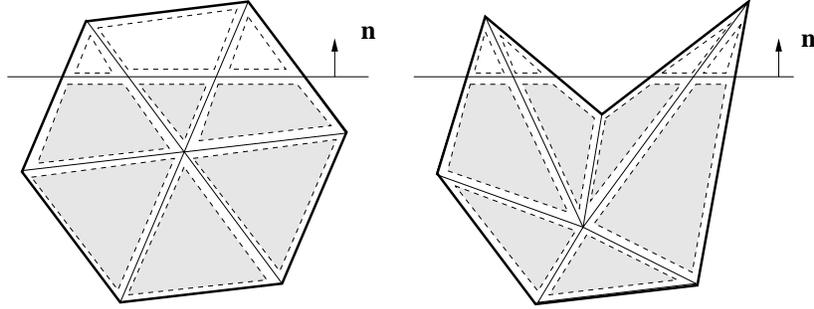


Figure 5: Two material mixed cell in 2D. Left represents for convex cell and right shows non-convex case. Notice that all sub-cells are always convex regardless of convexity of the original cell or presence of material interface.

generalized polyhedral mixed cells are represented in Fig. 6. Each cell includes three materials (red, green, and blue). Example of non-convex ennea-hedron, displayed in the middle row of Fig. 6, demonstrate case where one material (blue) is represented by disjoint pieces. Similar to 2D case each interface is reconstructed by intersection of the polyhedral cell with half-spaces. The sub-cells, initially tetrahedra, evolve to convex polyhedra as they intersect with corresponding half-spaces. The wire frame view of these sub-cells, shown in the left column of Fig. 6, reveals this sub-cell structures.

To summarize each material in mixed cell is represented as collection of convex polyhedrons. The hierarchical structure of a mixed cell is delineated in Fig. 7. Each of these polyhedrons were obtained as result of intersection of one of the original tetrahedra representing mixed cell with some half-spaces.

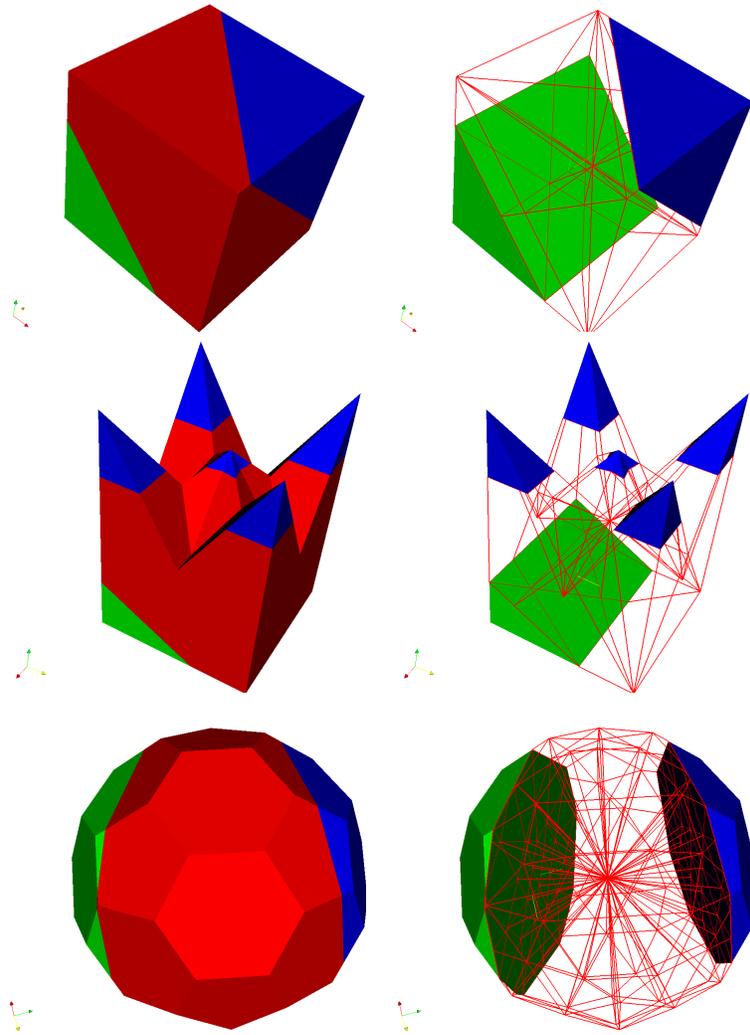


Figure 6: Generalized polyhedral cells with multi-materials (red, green, and blue). Top row shows hexahedral cell, middle row shows *non-convex* ennea-hedron (obtained by subdividing top face of hexahedron and disturbing the vertices on the faces), and bottom row represents truncated icosahedron. Left column show solid view, and right column show the wire-frame view of the solid which reveals the sub-cell structure.

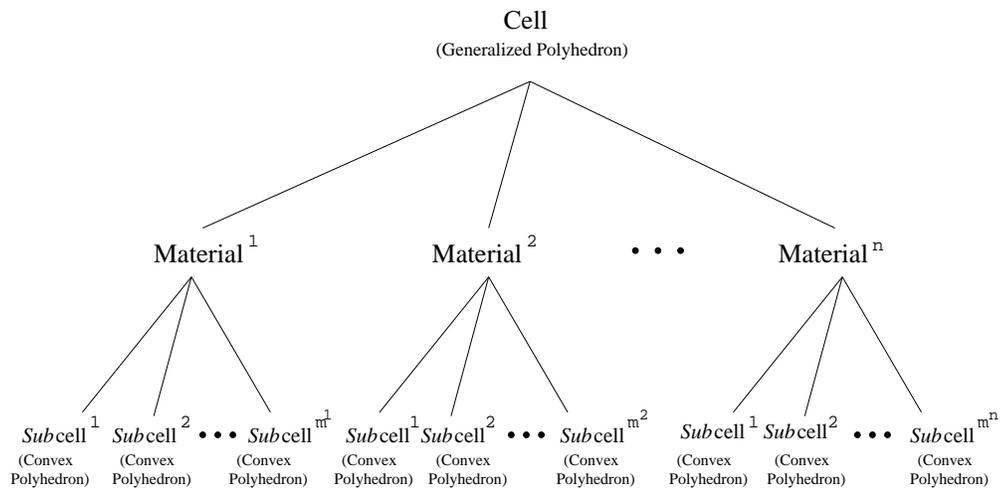


Figure 7: Hierarchical structure of generalized polyhedral cell (may be *non-convex*) containing multi-material. Each material has a set of non-overlapping *pure* sub-cells, which are always convex.

### 3 Interface reconstruction for two materials

Three representative piece-wise linear interface calculation (PLIC) methods are discussed in two material case, namely gradient based method, least squares volume-of-fluid interface reconstruction algorithm, and currently developed moment-of-fluid are presented. Next, numerical examples of interface reconstruction are demonstrated by using the three introduced methods, and their characteristics are compared. Mesh convergence study with spherical and cubic material regions is presented, and more complex geometry (character “A” shape) with sharp corners is tested. Finally, the spherical material region is reconstructed on unstructured tetrahedral mesh.

#### 3.1 Review of representative PLIC methods for case of two materials

In PLIC methods in each mixed cell interface between two materials is represented by plane. It is convenient to specify this plane in so-called *Hessian normal form*

$$\mathbf{n} \cdot \mathbf{r} + d = 0, \quad (3)$$

where  $\mathbf{r} = (x, y, z)$  point in the plane,  $\mathbf{n} = (n_x, n_y, n_z)$  are components of the unit normal to the plane, and  $d$  is the signed distance from the origin to the plane. The principal reconstruction constraint is local volume conservation, *i.e.* the reconstructed interface must truncate cell,  $c$ , with a volume equal to the reference volume  $V_c^{ref}$  of the material (or equivalently volume fraction  $f_c^{ref} = V_c^{ref}/V_c$ , where  $V_c$  volume of the entire cell  $c$ ).

Since a unique interface configuration does not exist, interface geometry must be inferred, based on local data and the assumptions of the particular algorithm. PLIC methods differ in how normal  $\mathbf{n}$  is computed. For given normal parameter  $d$  is uniquely defined from reference volume  $V_c^{ref}$ .

Most of standard PLIC methods,[45], use information only about volume fraction,  $f_c^{ref}$  in the cell  $c$ , under consideration and volume fractions,  $f_{c'}^{ref}$  in the neighboring cells  $c' \in \mathcal{C}(c)$  (here  $\mathcal{C}(c)$  set of cells which share face, edge, or vertex with cell  $c$ ). In our study standard PLIC methods are represented by two methods, which can be naturally extended to 3D mesh of generalized polyhedrons. The first method, which we will call GRAD, (see Section 3.1.1 for detail), is extension of *the central difference algorithm* and Parker and Youngs’ method as described in [45]. This method is based on estimate of gradient of volume fraction field to compute normal to interface. The second method, is extension of least squares volume-of-fluid interface reconstruction algorithm (LVIRA), [46, 45]. In LVIRA one is trying to find planar interface,

which being extended into cluster of surrounding cells gives the best (in least squares sense) match to the reference volume fractions in these cells under constraint that this plane exactly reproduces the volume fraction in the cell under consideration (see Section 3.1.2 for detail).

Both GRAD and LVIRA methods use only information about volume fractions.

The last method which we will study in this paper is so-called *moment-of-fluid* (MoF) method which was introduced recently, [14]. MoF method uses information about volume fraction and centroid, but *only* from the cell under consideration. Information from neighboring cells is not used. In MoF method the plane representing interface is chosen to match reference volume exactly and provide the best possible approximation (in least squares sense) to the centroid of the material (see Section [14] for detail).

In the rest of this Section we briefly describe main ideas of GRAD, LVIRA and MoF methods, give some details relevant to 3D extensions of these methods and present summary of algorithms needed for their implementation.

### 3.1.1 Gradient based interface reconstruction (GRAD)

In gradient based method, the interface normal,  $\mathbf{n}$ , is computed by approximating gradient of volume fraction function,  $f$ ,

$$\mathbf{n} \sim - \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right). \quad (4)$$

In case of 3D unstructured mesh consisting of generalized polyhedrons it is convenient to use least squares procedure (see, for example, [19]) to estimate gradient of volume fraction function. For each cell,  $c$ , linear function  $f_c(\mathbf{r})$ , can be reconstructed if following form

$$f_c(\mathbf{r}) = f_c(x, y, z) = f_c^{ref} + \sum_{\alpha=x,y,z} \left( \frac{\delta f}{\delta \alpha} \right)_c (\alpha - \alpha_c) \quad (5)$$

where  $f_c^{ref}$  is the reference volume fraction for cell- $c$ ,  $\left\{ \left( \frac{\delta f}{\delta \alpha} \right)_c, \alpha = x, y, z \right\}$  are estimates for components of volume fraction gradient,  $\alpha_c = \int_c \alpha dV / V_c$  are the coordinates of the cell centroids and  $V_c = \int_c 1 dV$  is the cell volume. The actual computation of the centroids and volume of polyhedral cell is based on the algorithm introduced in [38]. Estimates for components of the gradient can be found from minimization of the following functional

$$F \left( \frac{\delta f}{\delta x}, \frac{\delta f}{\delta y}, \frac{\delta f}{\delta z} \right) = \sum_{c' \in \mathcal{C}(c)} \left( f_{c'}^{ref} - \frac{\int_{c'} f_c(\mathbf{r}) dV}{V_{c'}} \right)^2 \quad (6)$$

Solution of least squares minimization problem reduces to solving  $3 \times 3$  system of linear equations.

Resulting normal  $\mathbf{n}$ ,

$$\mathbf{n} = - \left( \frac{\delta f}{\delta x}, \frac{\delta f}{\delta y}, \frac{\delta f}{\delta z} \right), \quad (7)$$

is taken as normal to interface in Eq. (3).

To complete interface reconstruction one need to find distance  $d$  in Eq. (3), such that intersection of corresponding half-space and cell  $c$  has volume  $V_c^{ref}$ .

For given normal  $\mathbf{n}$  half-space is uniquely defined by distance,  $d$  from the origin to the corresponding plane. Therefore, for given normal, volume of the intersection of the cell with half-space is function of  $d$ . Let us denote it by  $\mathcal{V}(d)$ . To find  $d$  we need to solve equation

$$\mathcal{V}(d) = V^{ref}. \quad (8)$$

Volume  $\mathcal{V}(d)$  is continuous and monotone function of  $d$ , which guarantees that equation always has unique solution. Procedure which we use in this paper to solve equation 8 is described in Section A.1 and it is based on secant and bisection methods. This procedure requires computation of volume of the intersection of the half-space and true polyhedron as basic operation, which is described in Section B. Let us note that all PLIC methods require solution of Eq. (8) (maybe several times), geometrical algorithms for intersection of the half-space and convex polyhedron, and computation of the volume of the polyhedron.

Detailed analysis of gradient based interface reconstruction methods in 2D is given in [45], where authors show that even on orthogonal mesh these methods are not capable to reconstruct linear interface exactly.

Therefore it is not surprise that GRAD method in 3D on GPM is incapable to recover planar interface as shown in Fig. 8. According to [45] it indicates its first-order accuracy of the GRAD method.

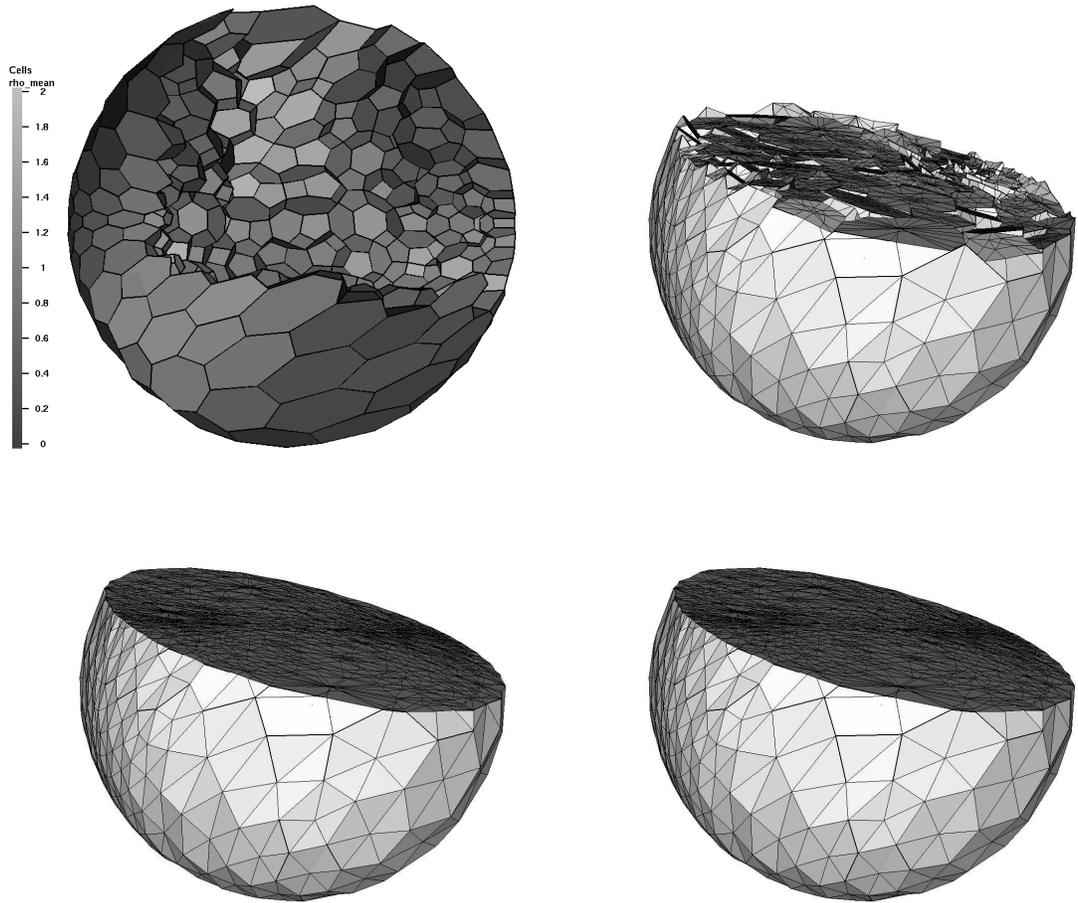


Figure 8: Interface reconstruction of the plane interface on a generalized polyhedral mesh. Top-left figure shows the original polyhedral mesh (subdivision of the cells into tetrahedra is not shown) obtained from [18]. Top-right figure shows GRAD reconstruction, which is first-order accurate method. Bottom-left figure shows LVIRA reconstruction and bottom-right figure shows MoF reconstruction, which are both planarity-preserving (second-order) accurate methods.

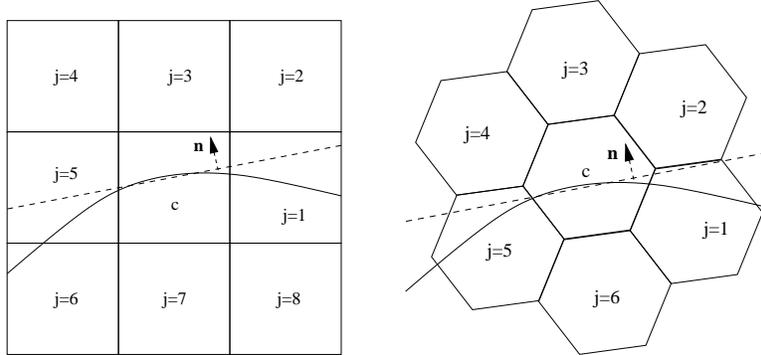


Figure 9: Stencil for LVIRA error computation in two dimensions. Stencil composed of immediate vertex neighbors of the central cell- $c$ . Solid curved line represents true interface, and dashed straight line represents the extension of piecewise linear volume fraction matching interface at the central cell- $c$ . Left picture represents for structured quadrilateral mesh, and right picture for unstructured polygonal mesh.

### 3.1.2 Least squares volume-of-fluid interface reconstruction algorithm (LVIRA)

In LVIRA interface reconstruction method, introduced by Puckett [46, 45], the interface normal is computed by minimizing the following error functional

$$E_c^{LVIRA}(\mathbf{n}) = \sum_{c' \in \mathcal{C}(c)} (f_{c'}^{ref} - f_{c'}(\mathbf{n}))^2 \quad (9)$$

where  $f_{c'}^{ref}$  is reference volume fraction of neighbor  $c'$ , and  $f_{c'}(\mathbf{n})$  is the actual (reconstructed) volume fraction of neighbor  $c'$  by extending the interface of central cell- $c$ , under constraint that corresponding plane exactly reproduces the volume fraction in the cell under consideration.

The stencil for the error computation of LVIRA is illustrated in Fig. 9, where 2D meshes are employed for simplicity. The neighboring cells around central cell- $c$  are referenced with index  $j$ . The stencil is composed of immediate vertex neighbors. Picture on the left of Fig. 9 represents structured quadrilateral mesh, and picture on the right shows stencil on unstructured polygonal mesh.

Like GRAD, LVIRA also requires information about volume fractions from all immediate neighboring cells.

Compare with GRAD method LVIRA requires minimization of non-linear objective function 9. In 3D normal can be described by polar angles, and therefore implementation of LVIRA requires algorithm for minimization of non-linear function of two variables. In this paper we use iterative Broyden-Fletcher-Goldfarb-Shanno (BFGS) method (see Appendix C for detail). Initial guess for normal is given by GRAD method. Implementation of this algorithm requires computation of  $E_c^{LVIRA}(\mathbf{n})$ . To do this for given normal  $\mathbf{n}$  we first need to find parameter  $d$  of the plane such that volume fraction in cell  $c$  exactly matches  $f_c^{ref}$ , which exactly the step which is performed in GRAD algorithm. Then resulting plane is extended into neighboring cells and intersection of the corresponding half-space with each neighboring cell  $c'$  is performed to compute actual volume fractions  $f_{c'}(n)$ . Next these volume fractions are used to compute error functional, expressed in Eq. (9). We will note that all these operations has to be performed on each iteration of minimization procedure.

With respect to GRAD method only one new algorithm is required — it is algorithm for non-linear optimization of function of two variables.

Clearly LVIRA method is much more expensive. For instance, in 3D logically Cartesian meshes, a central cell has 26 neighboring cells. Hence a single error function evaluation involves of 26 times polyhedral cell intersection on 3D hexahedral meshes.

In 2D LVIRA method is linearity-preserving, [45]. In 3D on GPM it is also planarity-preserving (second-order accurate), that it reconstructs planar interface exactly,(see, Fig. 8).

### 3.1.3 Moment-of-fluid interface reconstruction (MoF)

*Moment-of-Fluid* method was introduced by Dyadechko and Shashkov [14] for interface reconstruction in 2D. MoF method uses information about volume fraction,  $f_c^{ref}$  and centroid,  $\mathbf{x}_c^{ref}$  of the material, but *only* from the cell  $c$  under consideration. No information from neighboring cells is used.

In MoF method the interface representing interface is chosen to match reference volume exactly and provide the best possible approximation to the centroid of the material. That is, in MoF, the interface normal,  $\mathbf{n}$ , is computed by minimizing (under constraint that corresponding plane exactly reproduces the volume fraction in the cell under consideration) following functional

$$E_c^{MoF}(\mathbf{n}) = \|\mathbf{x}_c^{ref} - \mathbf{x}_c(\mathbf{n})\|^2 \quad (10)$$

where  $\mathbf{x}_c^{ref}$  is reference material centroid, and  $\mathbf{x}_c(\mathbf{n})$  is actual (reconstructed) material centroid with given interface normal  $\mathbf{n}$ .

Similar to LVIRA implementation of MoF method requires minimization of the non-linear function of two variables — we use the same BFGS algorithm as for LVIRA optimization. It is shown in [14] that vector pointing from reference centroid to centroid of entire cell is good initial guess for normal, and we use it in this paper.

Computation of  $E_c^{MoF}(\mathbf{n})$  requires following steps. First step is to find parameter  $d$  of the plane such that volume fraction in cell  $c$  exactly matches  $f_c^{ref}$ , this exactly the step which is performed in GRAD and LVIRA algorithms. Second, we need to compute centroid of the resulting polyhedron. This is simple calculation, which is described in [38]. Finally, one computes distance between actual and reference centroids.

Clearly from formal point of view MoF method is less expensive than LVIRA, because it does not require computation of any terms related to neighboring cells and computation of the centroids is relatively cheap. In real application, where computation of the reference centroids is a part of overall algorithm one will need to evaluate relative cost of different methods of interface reconstruction taking into account other considerations. This is beyond of scope of this paper.

In 2D MoF method is linearity-preserving, [14]. In 3D on GPM it is planarity-preserving (second-order accurate), that is, it reconstructs planar interface exactly,(see, Fig. 8).

## 3.2 Numerical examples

In this section, various examples of two material interface reconstruction is presented. We start with mesh convergence test is carried out for the accuracy checks of GRAD, LVIRA, and MoF reconstruction methods. Then examples of the interfaces with sharp corners and not-simply connected interfaces are also tested.

The initialization method of volume fraction and moment data is described in Appendix D.

### 3.2.1 Smooth interface

In this section we present convergence study for sing for GRAD, LVIRA and MoF interface reconstruction methods on example of reconstruction of spherical shape. The sphere is centered at  $(0.5 + \frac{1}{29}, 0.5 + \frac{1}{31}, 0.5 + \frac{1}{39})$  with radius  $r = 0.5 - \frac{1}{11}$ . Equispaced hexahedral meshes covering the domain of  $[0, 1]^3$  are used for the reconstruction.

In Fig. 10, we present results of the interface reconstruction using four levels of successively refined hexahedral meshes.

The first column of Fig. 10 shows the behavior of GRAD method. The gradient based scheme shows convergence to the true spherical interface as mesh refines, but the discontinuity of surface between cells persists even on the highest mesh resolution. This confirms that the gradient based scheme has interface normal error, which is independent of local mesh size,  $h$ . Hence the scheme is merely first-order accurate.

The second column displays LVIRA interface reconstruction. This method, formally second-order accurate, but demonstrates the worst performance on mesh of low resolution. This is because the interface is under-resolved and also indicates that the method is sensitive to the mesh resolution, however, with mesh refinement reconstructed interface looks more and more continuous. Even visually results obtained by LVIRA look better than one obtained by GRAD method.

In the last column interface reconstruction obtained by MoF method is presented. MoF interface reconstruction gives the best results for all levels of the mesh resolution.

The quantitative measures of the error in interface reconstruction is presented in Fig. 11

The error of interface reconstruction is measured by volume of symmetric difference defined as follows

$$T \triangle R = (T \cup R) - (T \cap R) \quad (11)$$

where  $T$  represents the set of true material region, which is sphere in this case, and  $R$  represents the set of actual (reconstructed) material region on a given mesh. Hence the expression in Eq. (11) concerns about *volume* of the material region not surface of the interface. The actual computation of the volume expressed in Eq. (11) is carried out in cell-wise as follows

$$\sum_{c \in \mathcal{M}} |T_c \triangle R_c| = \sum_{c \in \mathcal{M}} |(T_c \cup R_c) - (T_c \cap R_c)| \quad (12)$$

where  $\mathcal{M}$  is set of mixed cells,  $T_c$  and  $R_c$  are true and reconstructed material region within mixed cell- $c$ .  $|T_c \triangle R_c|$  represents the volume of the regions defined by  $T_c \triangle R_c$ .

The convergence rate of the actual reconstruction is displayed in Fig. 11. The results on Fig 11 demonstrate that: (i) gradient based scheme is only first-order accurate, (ii) LVIRA can be less accurate than GRAD on coarse mesh, but shows second-order accuracy on fine mesh, and (iii) MoF is second-order accurate and is most accurate for all levels of mesh resolution.

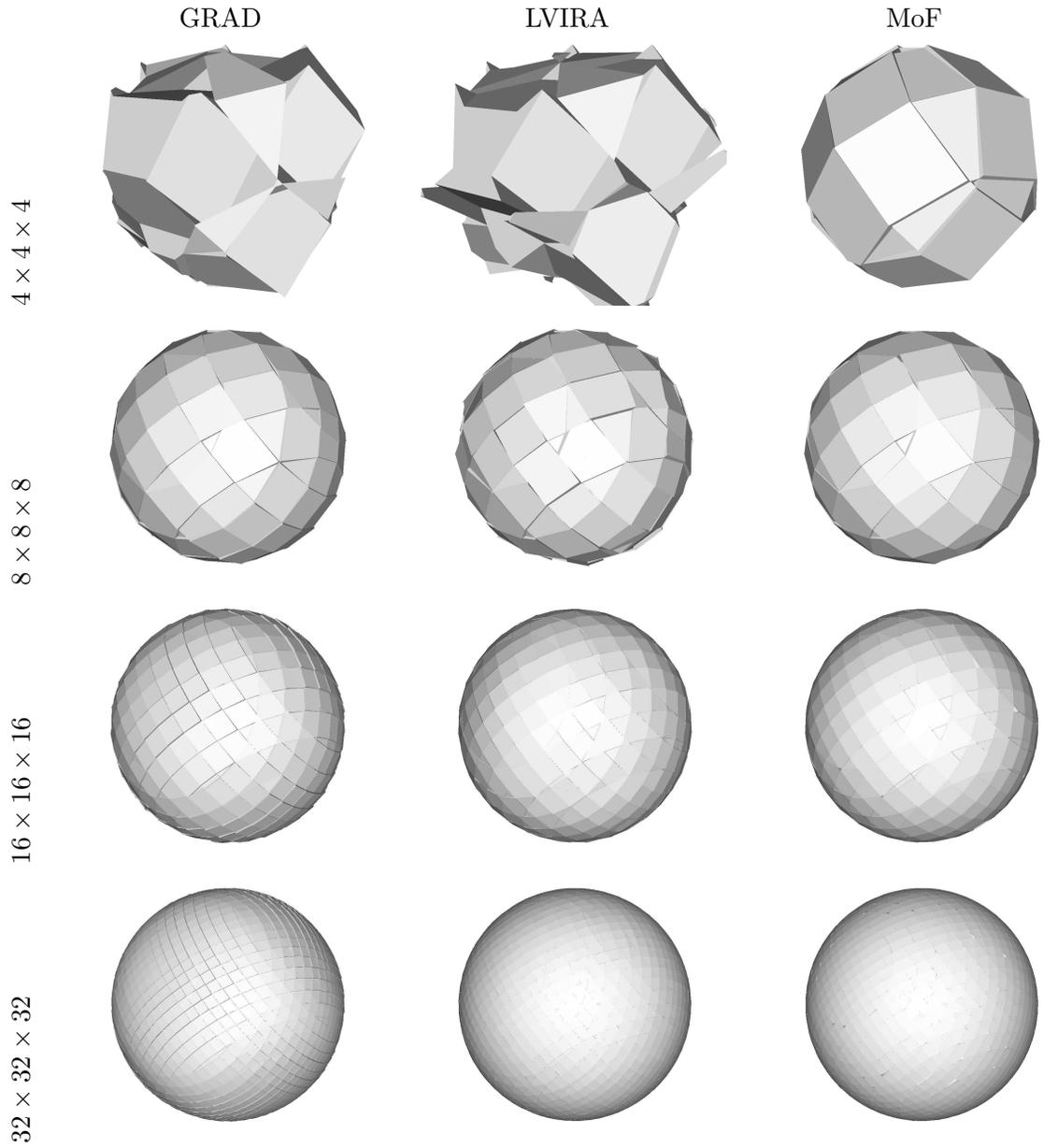


Figure 10: Mesh convergence study with spherical material region. Equispaced hexahedral meshes covering domain of  $[0, 1]^3$  are employed.

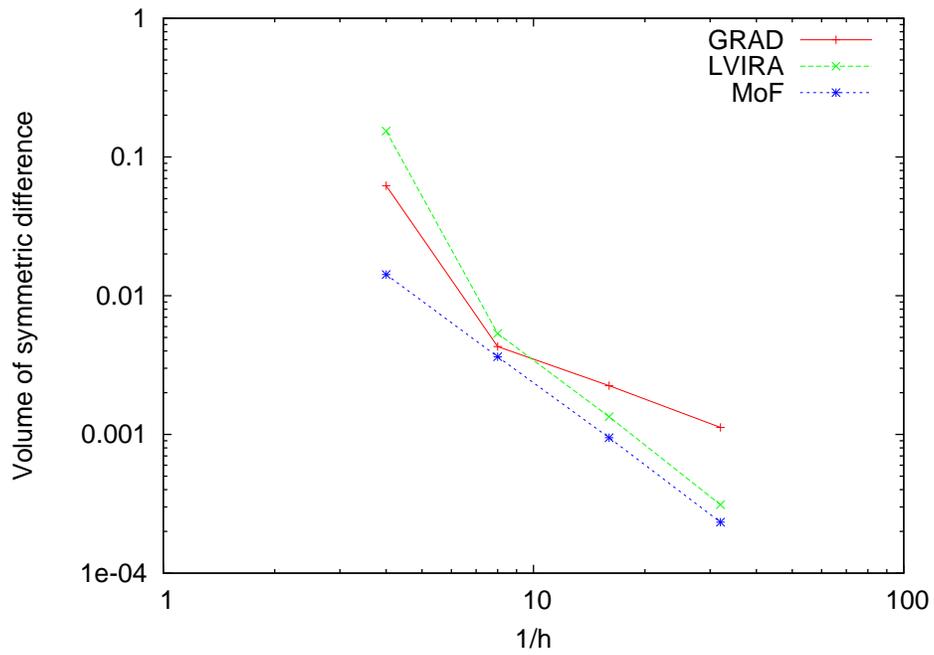


Figure 11: Convergence of reconstruction error measured by volume of symmetric difference between true and reconstructed material regions: spherical material shape. The convergence rates at the last refinement step are:  $p^{GRAD} = 0.999$ ,  $p^{LVIRA} = 2.107$ ,  $p^{MoF} = 2.019$

### 3.2.2 Interface with sharp corners

There are realistic situation where sharp interfaces are present and accurate reconstruction of such interfaces is important. For example sharp corners in a contact problems or triple points in multi-material ( $n_{mat} \geq 3$ ) flow simulations.

Here we present mesh convergence study for a cubical material shape. In order to define the cubical material region, first a bounding box is described by two vertices  $(0.21, 0.21, 0.21)$  and  $(0.81, 0.82, 0.81)$ . Then, the bounding box is rotated along the vector  $\mathbf{v} = \{1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3}\}$  with angle of  $\alpha = \pi/6$ . The actual reconstruction of the material region by using equispaced hexahedral meshes is displayed in Fig. 12. As also noted in Fig. 8, the first-order accurate GRAD is not capable of recovering flat interface exactly, but the second order accurate methods, LVIRA and MoF, preserve planarity. By the virtue of its local nature, the sharp edges of cube are best resolved in MoF

The error of interface reconstruction is measured by the volume of symmetric difference between the true and reconstructed material regions. The rate of convergence is displayed in Fig. 13. Both LVIRA and MoF shows second-order accuracy, but GRAD deviates towards first order accuracy as mesh refines. For all mesh resolutions, MoF shows the most accurate results.

As another test case of two material interface reconstruction, we present the interfaces with sharp corners including not simply connected material region shown in Fig. 14.

Fig. 14 illustrates the characteristics of the three interface reconstructions with sharp corners. The original surface mesh is obtained from [20]. First of all, the gradient based and LVIRA shows significant smoothing along the sharp corners. This is because their dependency on the neighbor cells, *i.e.* non-locality. Both methods needs information from neighbors either because of gradient computation or due to volume fraction error contribution. On the contrary, MoF reconstruction shows quite well resolved corners within the tolerance of single cell size by the virtue of its independence from the neighbors, *i.e.* locality.

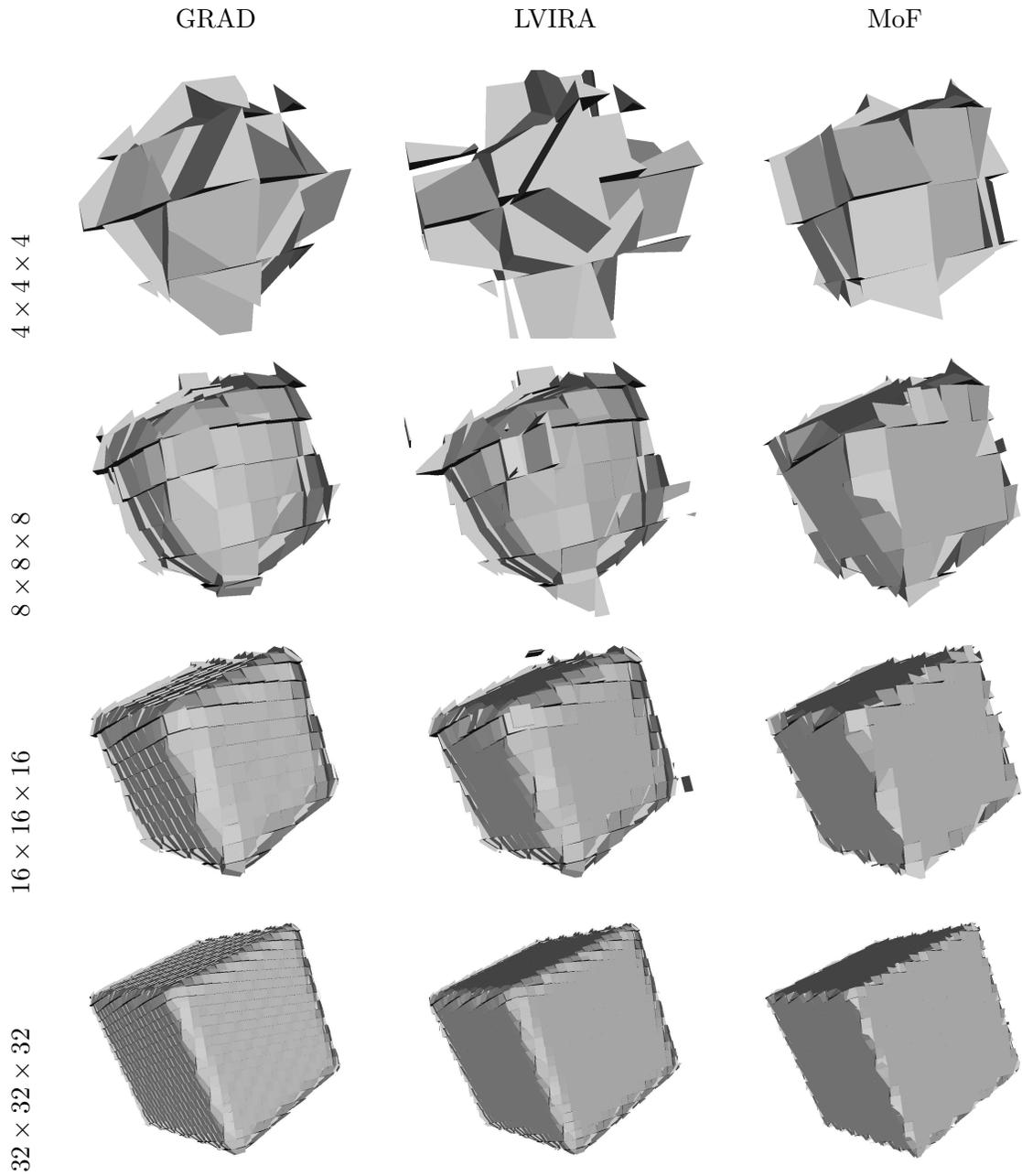


Figure 12: Mesh convergence study with cubical material region. Equispaced hexahedral meshes covering the domain of  $[0, 1]^3$  are employed.

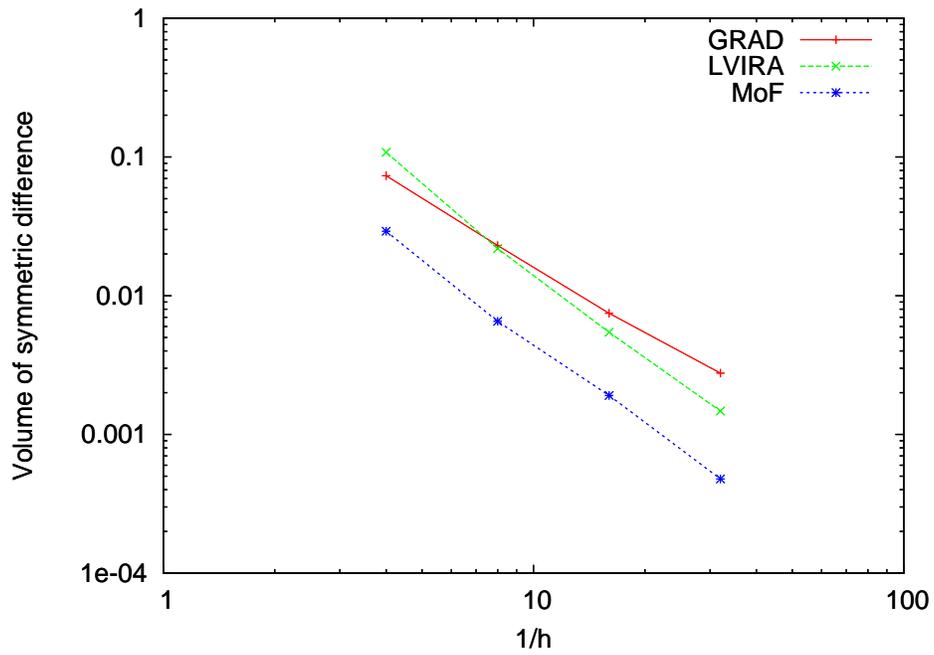
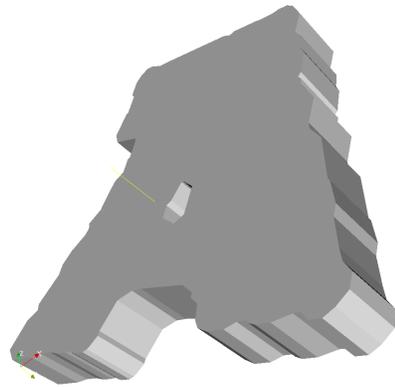
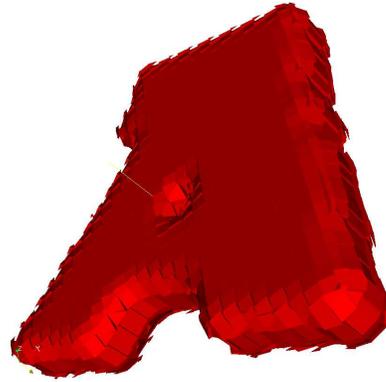


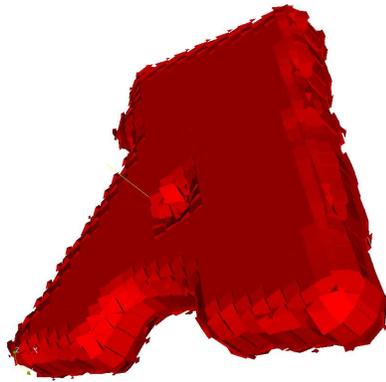
Figure 13: Convergence of reconstruction error measured augment. by volume of symmetric difference between true and reconstructed material regions: cubical material shape. The convergence rates at the last refinement step are:  $p^{GRAD} = 1.433$ ,  $p^{LVIRA} = 1.885$ ,  $p^{MoF} = 2.022$



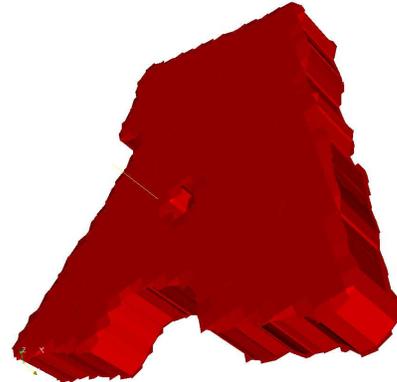
ORIGINAL



GRAD



LVIRA



MoF

Figure 14: Interface reconstruction of not simply connected material region with sharp corners.

### 3.2.3 Reconstruction on unstructured mesh

Interface reconstructions are demonstrated again on an unstructured tetrahedral mesh. A spherical material region centered at  $(0.5, 0.5, 0.5)$  with radius  $r = 0.4$  are considered. The unstructured tetrahedral mesh is generated with Gmsh [21]. The actual reconstruction of the spherical material region on the unstructured mesh is displayed in Fig. 15. The left-top figure shows subcell decomposed tetrahedral mesh, and top-right for GRAD, bottom-left for LVIRA, and bottom-right for MoF. The inter-cell interface discontinuity is minimal in MoF reconstruction, which strengthens the superiority of MoF method.

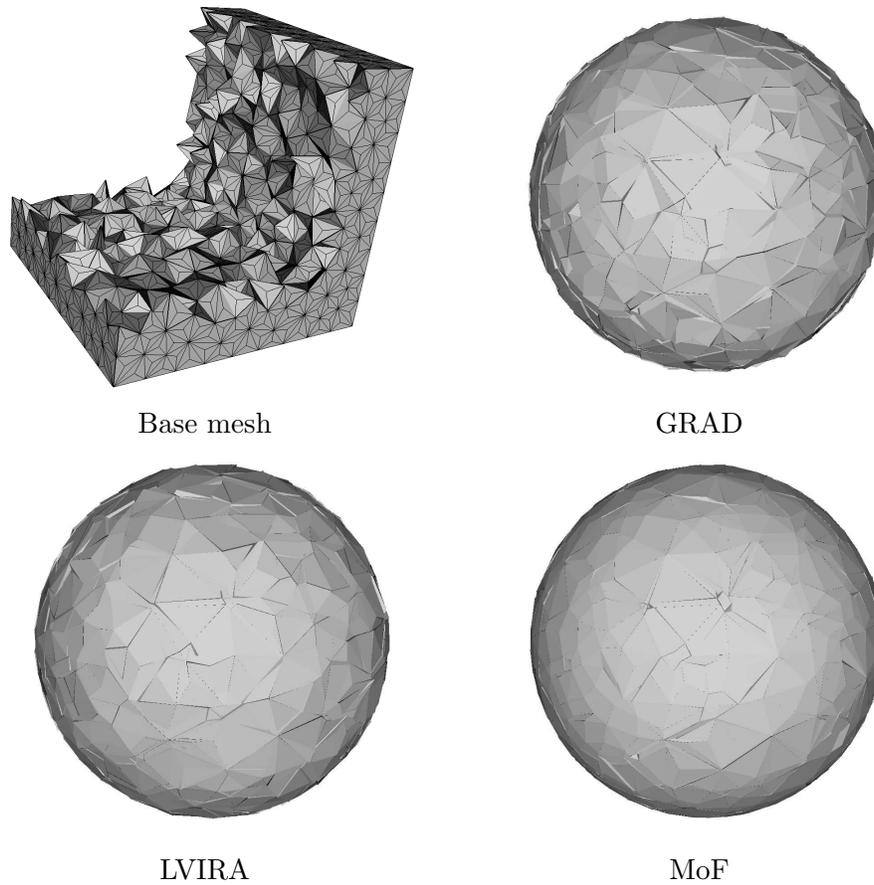


Figure 15: Two material spherical interface, centered at  $(0.5, 0.5, 0.5)$  with radius  $r = 0.4$ , reconstruction on unstructured tetrahedral mesh ( $n_{cells} = 8,659$ ). Top left – tetrahedral mesh (shown after subcell decomposition) covering the cubic domain of  $[0, 1]^3$ , Top right – GRAD reconstruction, bottom left – LVIRA reconstruction, and bottom right – MoF reconstruction.

## 4 Multi-material interface reconstruction

Example of the multi-material configuration is illustrated in Fig. 16. This four material configuration is produced using two intersecting eccentric spheres. Spheres ( $S_1$  and  $S_2$ ) have the same radius of  $r = 0.3$ , and centered at  $(0.4, 0.4, 0.4)$  and  $(0.6, 0.6, 0.6)$  respectively. Four different materials, *i.e.* green, red, blue, and gray regions, are described with two mutually intersecting spherical surfaces.

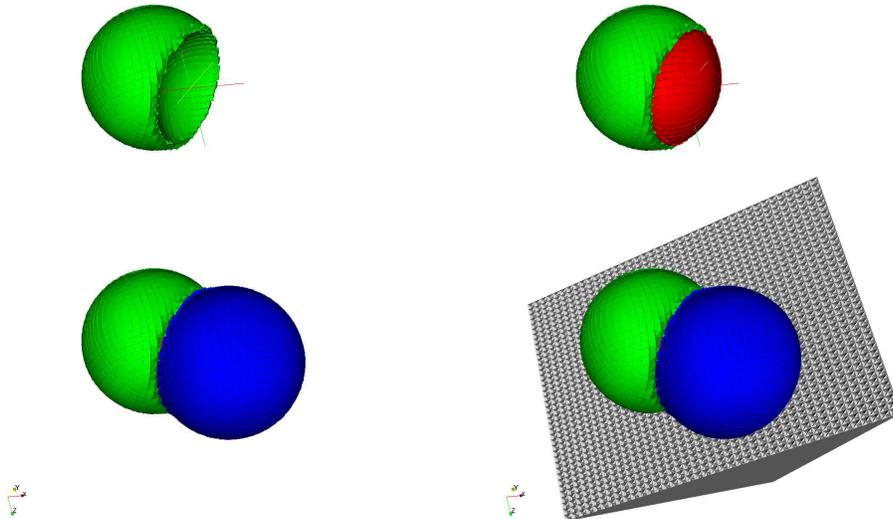


Figure 16: Example of multi-material configuration. Two intersecting eccentric spheres ( $S_1$  and  $S_2$ ) define four materials, namely green ( $S_1 - S_2$ ), red ( $S_1 \cap S_2$ ), blue ( $S_2 - S_1$ ), and gray (background) regions. Spheres have same radius of  $r = 0.3$ , and centered at  $(0.4, 0.4, 0.4)$  and  $(0.6, 0.6, 0.6)$  respectively.

Configuration presented in Fig. 16 will be used many times in this Section to demonstrate how different interface reconstruction methods work.

To explain logic of multi-material interface reconstruction we first will use 2D examples, which are easier to visualize. Also for simplicity of the visualization we will consider the case when 2D cell is convex polygon and therefore its subdivision into triangles is not needed. In Section 4.1 we will

explain logic of *nested dissection (ND)* strategy briefly described in introduction. In this Section we also describe the automatic choice of material ordering in framework of MoF method. In next Section 4.2 we will consider so-called *group nested dissection (GND)*, where nested dissection strategy is applied to several group of the materials. GND strategy allows to exactly reconstruct more complicated multi-material configuration.

Most of the presentation in this section is about MoF method, where normal to interface is defined as part of the optimization process. Hence, there is no ambiguity in normal computation of MoF method if material ordering is defined. In Section 4.4 we describe possible definitions of the interface normals for extension of the GRAD and LVIRA method to multi-material case.

Comparison of MoF, GRAD and LVIRA on some numerical examples is presented in Section 4.5.

## 4.1 Nested dissection

General idea of ND strategy is that materials are cut from mixed cell one by one in specified order as illustrated in Fig. 2. The ND strategy is illustrated with two representative cases of multi-material configuration, namely T junction and layered structures.

### 4.1.1 T-junction configuration

Let us demonstrate ND strategy on example of so-called T-junction configuration in 2D — Fig. 17.

In ND scheme one need to specify material ordering. Let us analyze what will happen if we choose material ordering as follows:  $(B, A, C)$  - which is optimal ordering for T-junction configuration. Let us note that in terminology used in introduction material #1 in material ordering is material  $A$ , material #2 =  $B$ , and material #3 =  $C$ .

According to ND scheme material #1 =  $A$  is separated from the rest of the materials in mixed cell. It is done using MoF method for two material case, where first material is material #1 =  $A$  and second material is union of materials #2 =  $A$  and #3 =  $C$ . That is for two material interface reconstruction on this step  $f_1 = f_B = 0.5$ , and  $f_{2,3} = f_A + f_C = 0.5$  and  $\mathbf{x}_c^1 = \mathbf{x}_c^B = (0.75, 0.5)$ ,  $\mathbf{x}_c^{2,3} = (0.25, 0.5)$ .

This first step creates pure sub-cell  $\Omega_1$ , as shown in Fig. 18 (a), which represent material #1 =  $A$ , and set  $\Omega \setminus \Omega_1$  which contains materials #2 =  $A$  and #3 =  $C$ . Next (and in case of three materials the last) step is to create

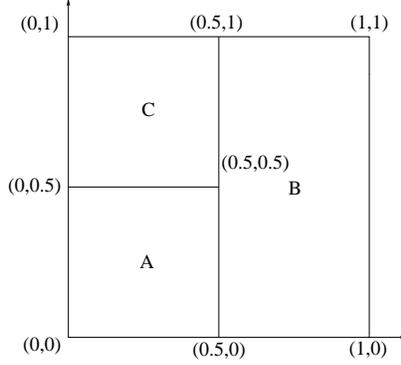


Figure 17: Three material T junction configuration. Entire cell  $\Omega$  is unit square. Materials  $A$  and  $C$  occupy left-half of the cell, and material  $B$  occupies right-half of the cell. Corresponding volume fractions and centroids are:  $f_A = 0.25$ ,  $f_B = 0.5$ ,  $f_C = 0.25$ ;  $\mathbf{x}_c^A = (0.25, 0.25)$ ,  $\mathbf{x}_c^B = (0.75, 0.5)$ ,  $\mathbf{x}_c^C = (0.25, 0.75)$

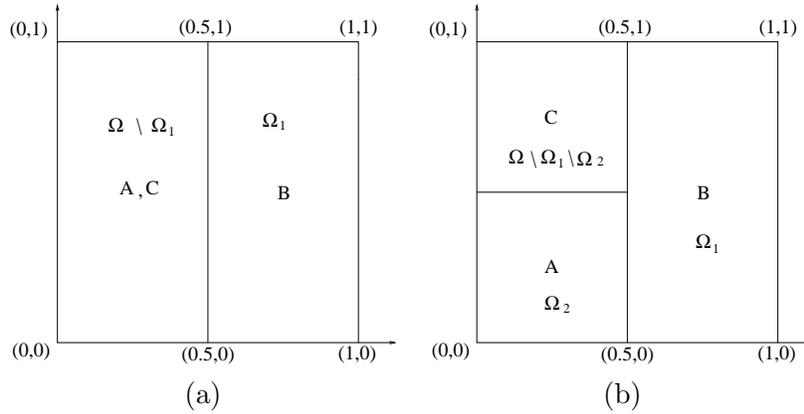


Figure 18: ND interface reconstruction of T junction –  $(B, A, C)$  ordering. (a) result of the first dissection, (b) result of the second (final in three material case) dissection.

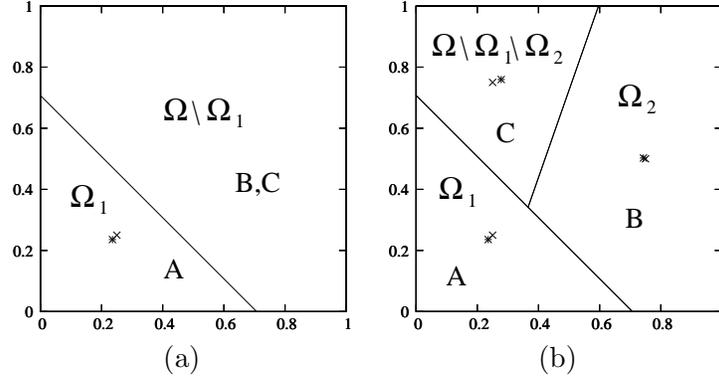


Figure 19: ND interface reconstruction of T junction –  $(A, B, C)$  ordering: (a) result of first dissection, (b) final result. Reference centroids are marked by  $\times$  and actual centroids of reconstructed pure sub-cell are marked by  $*$ .

pure sub-cell which represent material #2 =  $A$ . This again can be done by applying two material algorithm to  $\Omega \setminus \Omega_1$ , where first material will be material #2 =  $A$  and the second material will material #3 =  $C$ . Here one need to understand that second step is performed with respect to set  $\Omega \setminus \Omega_1$ , and therefore volume fractions for materials #2 =  $A$  and #3 =  $C$  has to be recomputed, and in our case they will be both equal to  $1/2$ . This second step, as shown in Fig. 18 (b), creates pure sub-cell representing material #2 =  $A$ ,  $\Omega_2$ . Finally, material #3 =  $C$  is represented by pure subcell  $\Omega \setminus \Omega_1 \setminus \Omega_2$ . For material ordering  $(B, A, C)$  we will recover original T-junction material configuration.

Let us note that in this case all centroids for all materials are reproduced exactly.

Now let us consider different ordering  $(A, B, C)$ . In this case analogs of Figures 18 with some additional information are presented in Fig. 19.

In true configuration (Fig. 17) material  $A$  is square, that is, its interface with rest of materials is broken line and cannot be represented by one line. For chosen material ordering  $(A, B, C)$  after first dissection we obtain sub-cell (triangle)  $\Omega_1$  (Fig. 19 (a)) which represents material #1 =  $A$ . The area of this triangle equals the area occupied by material  $A$  in true configuration, but centroid, marked by  $*$ , is different from reference centroid marked by  $\times$ . Final result of reconstruction is shown in Fig. 19 (b). It also shows positions of actual and reference centroids for the rest of the materials. For this material ordering total centroid discrepancy (error),  $E_{(A,B,C)}$ , in the

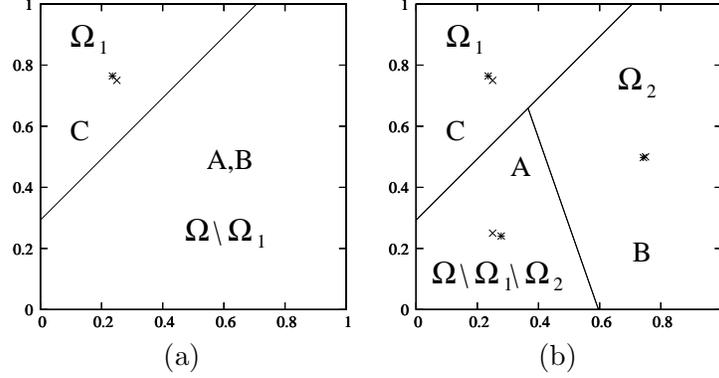


Figure 20: ND interface reconstruction of T junction –  $(C, B, A)$  ordering. (a) result of the first dissection, (b) result of the second (final) dissection. Reference centroids are indicated by  $\times$  and actual centroids are marked by  $*$ .

position of the centroids, defined by,

$$E_{(A,B,C)} = \sum_{m=1,2,3} \|\mathbf{x}_m^{ref} - \mathbf{x}_m^{act}\|^2,$$

equals to 0.0366461. There are four other material ordering are possible, but, because of symmetry of T junction configuration, only one –  $(C, B, A)$  produces distinct result presented in Fig. 20.

For this ordering total centroid discrepancy equals to 0.0366461. In multi-material extension of MoF method one need to try all possible ordering and then choose one with minimal total centroid discrepancy. In our case optimal material ordering will be  $(B, C, A)$  or  $(B, A, C)$ , for both of the discrepancy is zero. Here we need to note that for T-junction configuration it is enough to try only three different choices of first material and then compare discrepancy in position of its centroid to choose optimal material ordering.

#### 4.1.2 Layered configuration

Let us present another configuration where similar strategy will work. It is case of “layered” materials as delineated in Fig. 21. Entire cell  $\Omega$  is unit square. Material  $A$  occupies triangle with vertices:  $(0, 0.1)$ ,  $(0.3, 1)$ ,  $(0, 1)$ . Material  $B$  occupies pentagon with vertices:  $(0, 0.1)$ ,  $(0, 0)$ ,  $(0.5, 0)$ ,

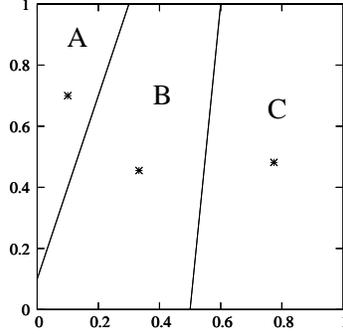


Figure 21: Three material *layered* configuration. Positions of reference centroids are marked by  $\times$  and actual centroids by  $*$ . The original configuration is reconstructed exactly with material orderings  $(A, B, C)$ ,  $(A, C, B)$ ,  $(C, B, A)$ , or  $(C, A, B)$ .

$(0.6, 1)$ ,  $(0.3, 1)$ . And finally, material  $C$  occupies quadrangle with vertices:  $(0.5, 0)$ ,  $(1, 0)$ ,  $(1, 1)$ ,  $(0.6, 1)$ . Corresponding volume fractions and centroids are:  $f_A = 0.135$ ,  $f_B = 0.415$ ,  $f_C = 0.45$ ;  $\mathbf{x}_c^A = (0.1, 0.7)$ ,  $\mathbf{x}_c^B = (0.332931726908, 0.455020080321)$ ,  $\mathbf{x}_c^C = (0.774074074074, 0.481481481481)$ .

Original configuration is reproduced exactly with any material ordering where material  $B$  is not on first position. Result of reconstruction for material ordering  $(B, C, A)$  presented in Fig. 22-(a). In this case discrepancy is equal to 0.384038. It is interesting to note that for this ordering reference centroid for material  $A$  located outside corresponding reconstructed pure sub-cell. This configuration also demonstrates that in general results of reconstruction when two last materials are interchanged are not the same. Result of reconstruction for material ordering  $(B, A, C)$  presented in Fig. 22-(b). In this case discrepancy is equal to 0.342561. Clearly in case of layered materials we can find optimal ordering similar to T-junction configuration by just comparing how well actual centroid of first material fits corresponding reference centroid.

### 4.1.3 Automatic ordering for ND

In search of best material ordering for nested dissection, two types of question has to be answered: first (i) local or global ordering, that is applying the same ordering for all cells or applying different ordering for each cell, and (ii) how to determine the best order. First issue of multi-material order-

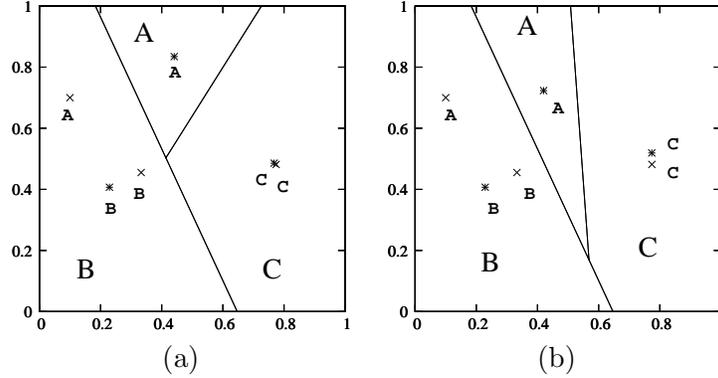


Figure 22: ND interface reconstruction of layers. (a) corresponds to material ordering  $(B, C, A)$ , reference ( $\times$ ) and actual ( $*$ ) centroids are also presented. These centroids have additionally marked by corresponding material index. (b) present results for material ordering  $(B, A, C)$ .

ing is the *scope* of ordering; whether a certain ordering applies globally or locally. It can be expected that the best material ordering would not be the same for every cell but have to change from cell to cell, *i.e.* *local* ordering. Next issue is about how to find the optimal order. The material orders may be assigned by the user, but ideally the optimal order would be better to be determined without user intervention, *i.e.* *automatic* ordering. Due to their non-local nature (dependence on neighbors), it is not clear how to extend GRAD and LVIRA reconstruction schemes to multi-material case with automatic-local ordering. However MoF, with its local nature, can achieve the automatic-local ordering.

Now the automatic ordering determination strategy is presented with our primary reconstruction scheme, MoF. The algorithm as presented in [15] is a straight forward extension from two material interface reconstruction. In two material case, the interface is determined such that deviation of actual centroid from the reference is minimized. The same strategy can be applied for multi-material interfaces reconstruction as well as proper ordering selection. For any given material order,  $\mathbf{m}$ , the total deviation of actual centroids from the reference centroids can be computed as follow

$$E(\mathbf{m}, \mathbf{N}) = \sum_{m_i^{id} \in \mathbf{m}} \|\mathbf{x}_c^{ref}(m_i^{id}) - \mathbf{x}_c^{act}(m_i^{id}, \mathbf{n}_i)\|^2 \quad (13)$$

where  $\mathbf{m} = (m_1^{id}, m_2^{id}, \dots, m_n^{id})$  is the ordering vector containing material-

id's,  $\mathbf{N} = (\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_{n-1})$  is ordered list of corresponding interface normals computed by MoF optimization,  $n$  is total number of materials inside of the cell,  $\mathbf{x}_c^{ref}(m_i^{id})$  is the reference centroid of material  $m_i^{id}$ , and  $\mathbf{x}_c^{act}(m_i^{id}, \mathbf{n}_i)$  is the actual centroid of material  $m_i^{id}$  reconstructed with interface normal  $\mathbf{n}_i$ . For cells with  $n$  number of materials, the number of all possible ordering ( $\mathbf{m}$ ) is  $n!$ . The optimal ordering,  $\mathbf{m}^*$ , comes with the least accumulated error in centroids as expressed in Eq. (13).

This way of searching optimal orders results in exhaustive test, *i.e.*  $n!$  evaluation of Eq. (13). This, however, leaves the door open to the possible short cuts. For example, part of ordering vector can be fixed for the materials of which the centroid error of reconstruction is below a certain tolerance. For the present purpose, introducing the idea of automatic local ordering, the exhaustive search is performed for cells containing more than two materials.

## 4.2 Group nested dissection

In this section we describe new *group nested dissection* (GND) method which is extension of nested dissection.

### 4.2.1 Motivation

As motivation for new method, let us consider "four corner" four material configuration (+ junction) presented in Fig. 23. Results of application of ND scheme for two possible material ordering is presented in Fig. 24.

From Fig. 24 it is clear that there is no material ordering in framework of ND scheme which will allow to reconstruct "four corner" (+ junction) configuration exactly. The reason for this is that, there is no material which internal can be represented as segment of straight line.

### 4.2.2 GND

Let us revisit ND scheme. Two main components of this scheme is that on each step we use algorithm for two materials, and each time we dissect some piece of the cell. It is clear that instead of dissecting first from the rest as it is done in ND scheme we can dissect any *group* of materials from the rest. In case of "four corner" configuration one can choose first group  $G_1 = [A, D]$  and second group  $G_2 = [B, C]$ . With volume fractions  $f_{G_1} = f_{G_2} = 0.5$  and centroids  $\mathbf{x}^{G_1} = (0.25, 0.5)$ ,  $\mathbf{x}^{G_2} = (0.75, 0.5)$ . Then the first step of *group nested dissection* (GND) will be to dissect group  $G_1$  from group  $G_2$ . It is clear that both groups will be reconstructed exactly. After this we introduce sub-groups  $G_{1,1} = A$ ,  $G_{1,2} = D$ , and sub-groups  $G_{2,1} = B$ ,  $G_{2,2} =$

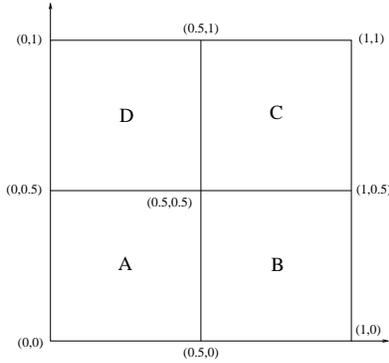


Figure 23: Four material - “four corner” (+ junction) configuration. Entire cell  $\Omega$  is unit square. Each of materials occupies one of the quadrant of the cell. Corresponding volume fractions and centroids are:  $f_A = f_B = f_C = f_D = 0.25$ ;  $\mathbf{x}_c^A = (0.25, 0.25)$ ,  $\mathbf{x}_c^B = (0.75, 0.25)$ ,  $\mathbf{x}_c^C = (0.5, 0.75)$ ,  $\mathbf{x}_c^D = (0.75, 0.75)$ .

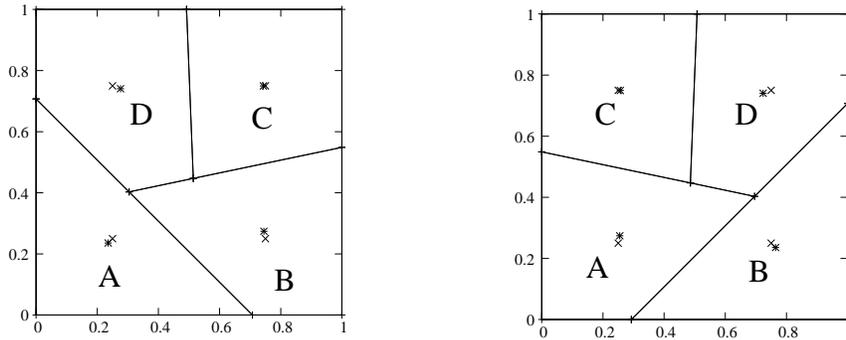


Figure 24: ”Four corner” configuration. Interface reconstruction using nested dissection scheme: left -  $(A, B, C, D)$  ordering, right -  $(B, A, C, D)$  ordering. Positions of reference,  $\times$ , and actual centroids,  $*$ , are also presented.

$C$ , and apply GND (which will now be the same as ND) for each group dissecting them into corresponding subgroups. This algorithm reproduces "four corner" (+ junction) configuration exactly.

Let us note that ND strategy with material ordering  $(A, B, C, D)$  can be considered as GND with  $G_1 = A$ ,  $G_2 = [B, C, D]$ , then in group  $G_2$  one introduces sub-groups  $G_{2,1} = B$ ,  $G_{2,2} = [C, D]$ , and finally  $G_{2,2,1} = C$ ,  $G_{2,2,2} = D$ .

In general, in GND scheme, role of the single material in ND scheme is played by group of materials, and then in each group GND scheme applied again until sub-groups will consist of one material. For example, for configuration presented in Fig. 25 the GND scheme, which reproduces it exactly, looks like this:  $G_1 = [A, E]$ ,  $(G_{1,1} = A, G_{1,2} = E)$ ;  $G_2 = [B, C, D]$ ,  $(G_{2,1} = B, G_{2,2} = [C, D], (G_{2,2,1} = C, G_{2,2,2} = D))$ .

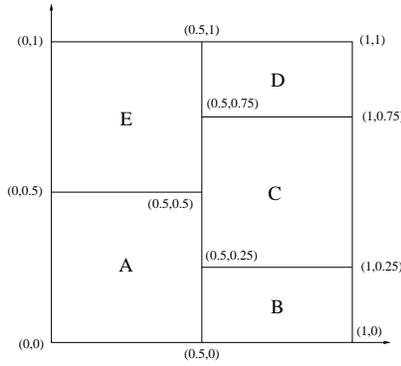


Figure 25: Five materials- "double T-junction" configuration. Entire cell  $\Omega$  is unit square. Corresponding volume fractions and centroids are:  $f_A = 0.25$ ,  $f_B = 0.125$ ,  $f_C = 0.25$ ,  $f_D = 0.125$ ,  $f_E = 0.25$ ;  $\mathbf{x}_c^A = (0.25, 0.25)$ ,  $\mathbf{x}_c^B = (0.75, 0.125)$ ,  $\mathbf{x}_c^C = (0.75, 0.5)$ ,  $\mathbf{x}_c^D = (0.75, 0.875)$ ,  $\mathbf{x}_c^E = (0.25, 0.75)$ .

For GND scheme optimal choice of groups and their ordering is defined exactly as for ND scheme that is one which gives smallest discrepancy in positions of the centroids for all materials.

### 4.2.3 Automatic ordering for GND

In this section we give more formal description of ideas described in previous section. If a cell contains four or more materials, the possibility of interface

formation gets more complex. For examples as delineated in Fig. 26, a cell with quadruple line or multiple triple lines, it can produce group structure inside of a cell. In such cases, ND strategy fails to recover the correct interfaces, and intersecting a group of material (GND) should be considered for possible candidate of optimal ordering the cells with  $n_{mat} \geq 4$ .

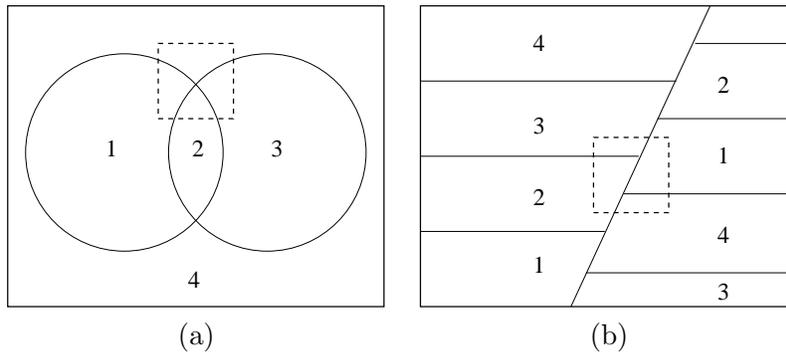


Figure 26: Interface structures of grouped multi-materials in 2D. (a) shows cell containing quadruple point and (b) shows multiple triple point in a cell (indicated with dashed lines). Multiple triple points (b) can be removed by mesh refinement, but quadruple point (a) persists regardless of mesh resolution.

The accumulated moment error by any possible GND scheme, described by the combination of material ordering ( $\mathbf{m}$ ) with grouping ( $\mathbf{g}$ ), can be expressed as follows.

$$E(\mathbf{m}, \mathbf{g}, \mathbf{N}) = \sum_{m_i^{id} \in \mathbf{m}} \|\mathbf{x}_c^{ref}(m_i^{id}, \mathbf{g}) - \mathbf{x}_c^{act}(m_i^{id}, \mathbf{g}, \mathbf{n}_i)\|^2 \quad (14)$$

where is the grouping vector  $\mathbf{g} = (g_1, g_2, \dots, g_{n-1})$  indicates cutting position within a given material order  $\mathbf{m} = (m_1^{id}, m_2^{id}, \dots, m_n^{id})$  and other notations are same as in Eq. (13). Each component  $g_i$  of grouping vector uniquely indicates the cutting position (*e.g.* if  $g_1 = 2$ , then first cut after second material in the ordering vector) within the given ordering vector. The nested

dissection strategy expressed in Eq. (13) can be interpreted as a special case group-nested dissection of Eq. (14) with grouping vector of  $\mathbf{g} = (1, 2, \dots, n - 1)$ . which means that the cutting position is sequential along the order of ordering vector.

In four material case, all possible number of sequential material ordering is  $4!$ . For any given material ordering, say  $\mathbf{m} = (2, 3, 1, 4)$ , possible grouping is only one because the result of grouping  $\mathbf{g} = (2, 1, 3)$

$$\{\mathbf{m} : \mathbf{g}\} = ( \underbrace{2, |^{cut^2} 3,}_{first\ group} |^{cut^1} \underbrace{1, |^{cut^3} 4}_{second\ group} )$$

will be identical to result of  $\mathbf{g}' = (2, 3, 1)$

$$\{\mathbf{m} : \mathbf{g}'\} = ( \underbrace{2, |^{cut^3} 3,}_{first\ group} |^{cut^1} \underbrace{1, |^{cut^2} 4}_{second\ group} ).$$

As a demonstration of the algorithm, however all possible grouping (even including nested dissection) can be expressed with Eq. (14), namely  $(n - 1)!$  grouping for any given ordering vector of size  $n$ .

As the number of material grows ( $n_{mat} \geq 6$ ) the way of exhaustive search can also express multi-layered grouping, *i.e.* group of groups. For example, in a cell containing seven material as displayed in Fig. 27, multi-layered grouping described by ordering vector  $\mathbf{m} = (1, 2, 3, 4, 5, 6, 7)$  and grouping vector  $\mathbf{g} = (3, 1, 2, 5, 4, 6)$  will result in

$$\{\mathbf{m} : \mathbf{g}\} = ( \underbrace{1, |^{cut^2} 2, |^{cut^3} 3,}_{first\ group} |^{cut^1} \underbrace{ \overbrace{4, |^{cut^5} 5,}^{first\ sub-group} |^{cut^4} \overbrace{6, |^{cut^6} 7}^{second\ sub-group} }_{second\ group} ) .$$

An example of multi-layered grouping is illustrated in Fig. 27. After all possible combinations (total  $n!(n - 1)!$  cases allowing redundancy) between ordering and grouping, the optimal interface reconstruction procedure can be determined regardless of the number of materials.

Once the ordering and grouping procedure is determined, the mechanism of multi-material interface reconstruction algorithm is identical to the two material case. In *nested* dissection strategy, interfaces are reconstructed in a certain order (either prescribed or determined on-the-fly) by separating the first material from the rest, second material from the rest, and so on. In *group-nested* dissection as illustrated in Fig. 28, interfaces are reconstructed by separating each material group from the rest, until all group is composed

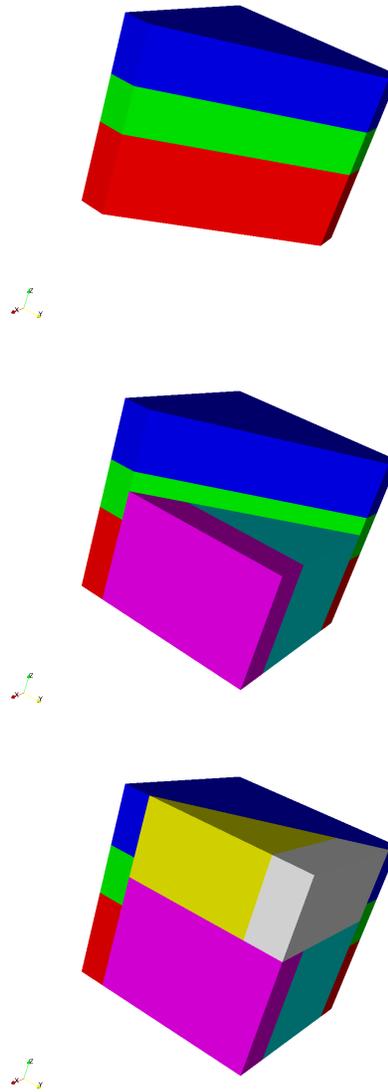


Figure 27: Multi-layered material grouping with a cell containing seven materials. From the top, groups of materials are added in the order of: the first group of three materials (red, green and blue), the first sub-group of the second group (cyan and magenta), and the second sub-group of the second group (yellow and grey).

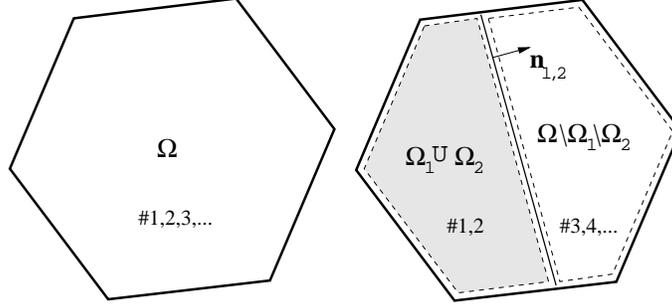


Figure 28: Group-nested dissection on 2D polygonal cell. Left shows initial configuration of multi-material cell, and right shows a group-nested dissection of the first group composed of materials #1 and 2.

of single material. Hence, nested dissection can be considered as a special case of group-nested dissection with all uni-material groups. The difference in intersection algorithm by considering grouping is that the input data for intersection should be the combination of materials contained in the group. Hence, volume fraction and centroid data for intersecting the material also have to be combined together in order to represent the group correctly. For example, a grouped intersection of  $n$  materials  $(1, 2, \dots, n)$  will need the following combination of volume fraction

$$f_G = \sum_{i \in \mathcal{S}(G)} f_i$$

and centroid data

$$\mathbf{x}_G = \sum_{i \in \mathcal{S}(G)} w_i \mathbf{x}_i$$

where  $\mathcal{S}(G)$  is the set of materials contained in group- $G$ ,  $\mathbf{x}_i$  is centroid of material- $i$  and  $w_i = f_i/f_G$  is weight for material- $i$ .

### 4.3 Material Ordering for GRAD and LVIRA

As we already mentioned, for MoF method optimal material grouping and ordering of the groups can be determined by first trying all possible ordering and groups and then choosing one which has smallest value of discrepancy in the positions of reference and actual centroids. This is exactly what we do in all MoF calculations presented in this paper. By design MoF method

is applied to each mixed cell independently and therefore material ordering inherently local, that it can be different for each mixed cell.

Choice of material ordering for traditional PLIC methods is unsolved problem. There were several attempts to develop algorithms automatic local schemes for material ordering [47, 39, 4, 1]. All of them are based on using information about exact or approximate of centroid position of the materials in mixed cell itself and its neighbors. To best of our knowledge, (see, e.g, examples in [15], and also comments in [39, 4]), none of these methods can treat basic cases (T junction, layers) correctly even on regular mesh. Existing ordering schemes have been described only in 2D. We do not want to speculate how these methods can be extended to 3D. In this paper, material ordering for GRAD and LVIRA methods will be explicitly specified for each example.

#### 4.4 Computation of the normals for GRAD and LVIRA in multi-material case

There are two approaches to compute normals for multi-material case for traditional PLIC methods, [3]. Both of these approaches based on basic premise that computation of the normals is always reducible to a two material case. The first approach is to calculate the interface between each material and the mixture of all other material in completely independent manner. We will not consider this approach in this paper because it is not used very often in practice. The second approach, which is called *onion skin*, [64, 5, 62, 10, 58] model is used much more often and we will describe it here (we need to note that in all these papers onion skin model is refers to interface reconstruction method for multi-material case and not just to computation of normals).

In onion skin model for cell containing  $n$  materials, with prescribed material ordering, the interface between material  $k$  and  $k + 1$  is defined as the interface between two materials one of which is mixture of materials  $1, 2, \dots, k$  and another one is mixture of the materials  $k + 1, k + 2, \dots, n$ . In onion skin model interface reconstruction and position can the be determined by techniques developed for two material configuration. The onion skin model is demonstrated in Fig. 29 for case of three materials.

The onion skin model was originally developed for layered materials. Left picture in Fig 29 presents valid interface reconstruction for case of layered materials. Picture on the right in Fig. 29 presents situation which can happen in practical application, when interfaces constructed using onion skin model intersect each other. This problem is well known [5, 10, 58]. For

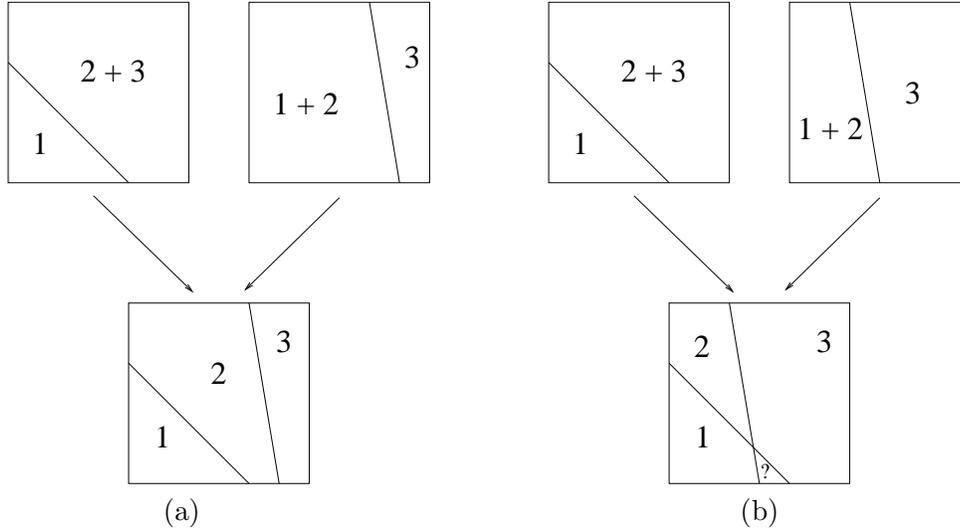


Figure 29: Onion skin interface reconstruction: (a) valid interface reconstruction, (b) invalid interface reconstruction with intersecting interfaces.

our purposes interface reconstruction method which do not always produces pure sub-cells for each material is not unacceptable.

Now we describe how to combine ND (or GND) scheme and ideas for computation of the normals taken from onion skin model in one algorithm. First we compute direction of the normal exactly as in onion skin model. This is only part of onion skin algorithm we will use. Let us note that computation of the normals can be done using different material ordering for each mixed cell.

According to ND scheme, which is illustrated in Fig. 2, we use normal between material #1 and the rest of the materials (computed as in onion skin model) to construct pure sub-cell containing this material ( $\Omega_1$  in Fig. 2). Now domain  $\Omega \setminus \Omega_1$  contains materials #2, #3, ..., #n. To create pure sub-cell for material #2. To do this we use onion skin normal computed for mixture of materials #1, #2 and mixture of the materials #3, #4, ..., #n. This normal is used to cut pure sub-cell  $\Omega_2$  from  $\Omega \setminus \Omega_1$ . This process is repeated until pure sub-cells are constructed for all materials.

The difference between onion skin interface reconstruction and ND interface reconstruction with onion skin normals is demonstrated in Fig. 30 for case of three materials.

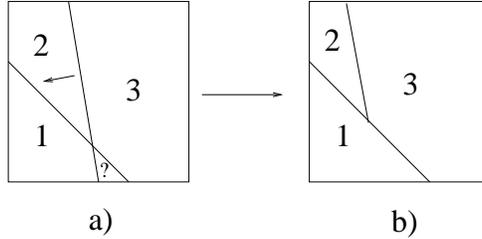


Figure 30: Comparison of onion skin interface reconstruction - left, and ND interface reconstruction with onion skin normals.

In left picture in Fig. 30 we repeat result of onion skin interface reconstruction from Fig. 29-(b), where interfaces are intersect each other. In right picture, which correspond to ND scheme with the same set of normals, position of the interface computed using second normal is adjusted (moved to the left) such that material #2 has specified volume.

Let us note again that onion skin normals are always computed for two "materials", each of them being mixture of original materials according to chosen material ordering. Therefore for GRAD and LVIRA computation of these normals will be done using corresponding algorithms for two materials described in Section 3.

In case of GND scheme each group is considered as material for purpose of computation of the corresponding normal.

#### 4.5 Numerical examples

In this Section we present examples of multi-material interface reconstruction. MoF with automatic local ordering strategy will be employed as the primary reconstruction method. LVIRA and GRAD, with prescribed material ordering, are also compared with MoF.

#### 4.6 Mesh convergence study

Mesh convergence study is presented for multi-material configuration. The three-material configuration, shown in Fig. 31, is utilized. For the visualization purpose, only fraction of the background (grey), which is originally surrounding the other materials (cyan and magenta), is displayed. In order to exclude any possible effect of ordering, all three PLIC methods presented are applied with same prescribed global ordering (cyan  $\rightarrow$  magenta  $\rightarrow$  grey).

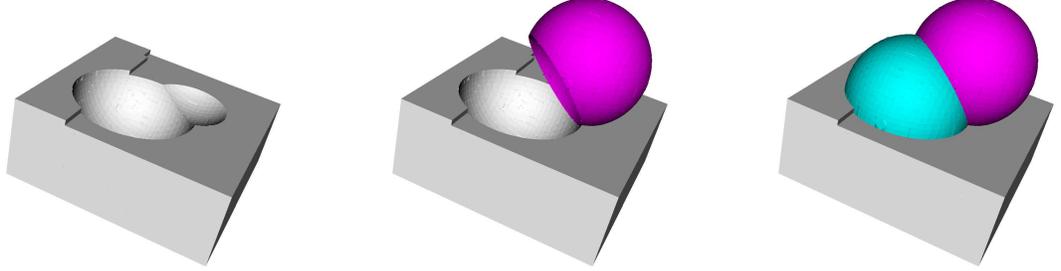


Figure 31: Multi-material configuration for mesh convergence study ( $n_{mat} = 3$ ). The configuration is similar as the one shown in Fig. 16. Two intersecting eccentric spheres ( $S_1$  and  $S_2$ , centered at  $(0.4, 0.4, 0.4)$  and  $(0.6, 0.6, 0.6)$  respectively, both with radius  $r = 0.3$ ) define three materials, namely material #1 defined by  $S_1$  (cyan), #2 by  $S_2 - S_1$  (magenta), and #3 by the background (grey). From the left, for visualization purpose fraction of #3, entire #2 and #1 are displayed accumulatively.

The reconstruction error is measured by the volume of symmetric difference between true and reconstructed material regions. Similar to the one in two material case in Eq. (11), the error of multi-material reconstruction is measured as follows

$$\sum_{j \in \mathbf{m}_{interior}} (T \Delta R)_j = \sum_{j \in \mathbf{m}} ((T \cup R)_j - (T \cap R)_j) \quad (15)$$

where  $\mathbf{m}_{interior} = \{1, 2, \dots, n_{mat} - 1\}$  is set of material except the background (for the example in Fig. 31,  $\mathbf{m}_{interior} = \{1, 2\}$ ), and  $T$  and  $R$  are the sets of true and reconstruction material regions.

The actual reconstruction of the multi-material configuration is displayed in Fig. 32. MoF shows the best reconstruction for all range of meshes. LVIRA shows second most accurate result on fine meshes. GRAD shows better result than LVIRA in low resolution, but the least accurate result as mesh refines. Notice that along the triple lines (where the two spheres and background are merging), only MoF reconstructs the triple lines with sharp angle but LVIRA and GRAD smears the angle near the triple lines.

The rate of convergence is displayed on Fig. 33. The overall behavior is in accordance with the result of two material case as shown in Fig. 11. In general, MoF is the most accurate method with second-order accuracy, and followed by another second-order accurate method, LVIRA, and GRAD

shows the least accurate result, which is first order.

#### 4.6.1 Computational cost

The overall computational cost of the reconstruction methods are measured in CPU time in Table 1. The machine with Opteron 2 GHz with 24 GB memory running 64 bit Fedora Core 3 is used for this benchmark test. Since the implementation of each method can be slightly different depending on source level optimization, this result report only rough estimate of each methods.

Mesh resolution	GRAD	LVIRA	MoF
$10 \times 10 \times 10$	4.81 <sup>s</sup>	52.70 <sup>s</sup>	36.90 <sup>s</sup>
$20 \times 20 \times 20$	24.80 <sup>s</sup>	5 <sup>m</sup> 25.00 <sup>s</sup>	3 <sup>m</sup> 08.00 <sup>s</sup>
$40 \times 40 \times 40$	2 <sup>m</sup> 28.00 <sup>s</sup>	22 <sup>m</sup> 27.00 <sup>s</sup>	16 <sup>m</sup> 36.00 <sup>s</sup>

Table 1: CPU time required for the reconstruction of three-material configuration displayed in Fig. 32.

Regardless of mesh resolution, GRAD is the fastest method and LVIRA is the most expensive method, MoF is in the middle. At the finest level of mesh ( $40 \times 40 \times 40$ ), MoF takes 6.7 times CPU time than GRAD, and LVIRA takes 9.1 times. The overall behavior of CPU cost with respect to the mesh size is displayed in Fig. 34. It is clear that all three methods shows linear complexity with respect to the mesh resolution. At each step of mesh refinement, approximately the number of mixed cells quadruples and so the CPU time, *i.e.*  $O(n)$  complexity, where  $n$  is number of mixed cells.

For large scale simulations, parallelization is straightforward in MoF by the virtue of its locality, *i.e.* no inter-cell communication is necessary for interface reconstruction.

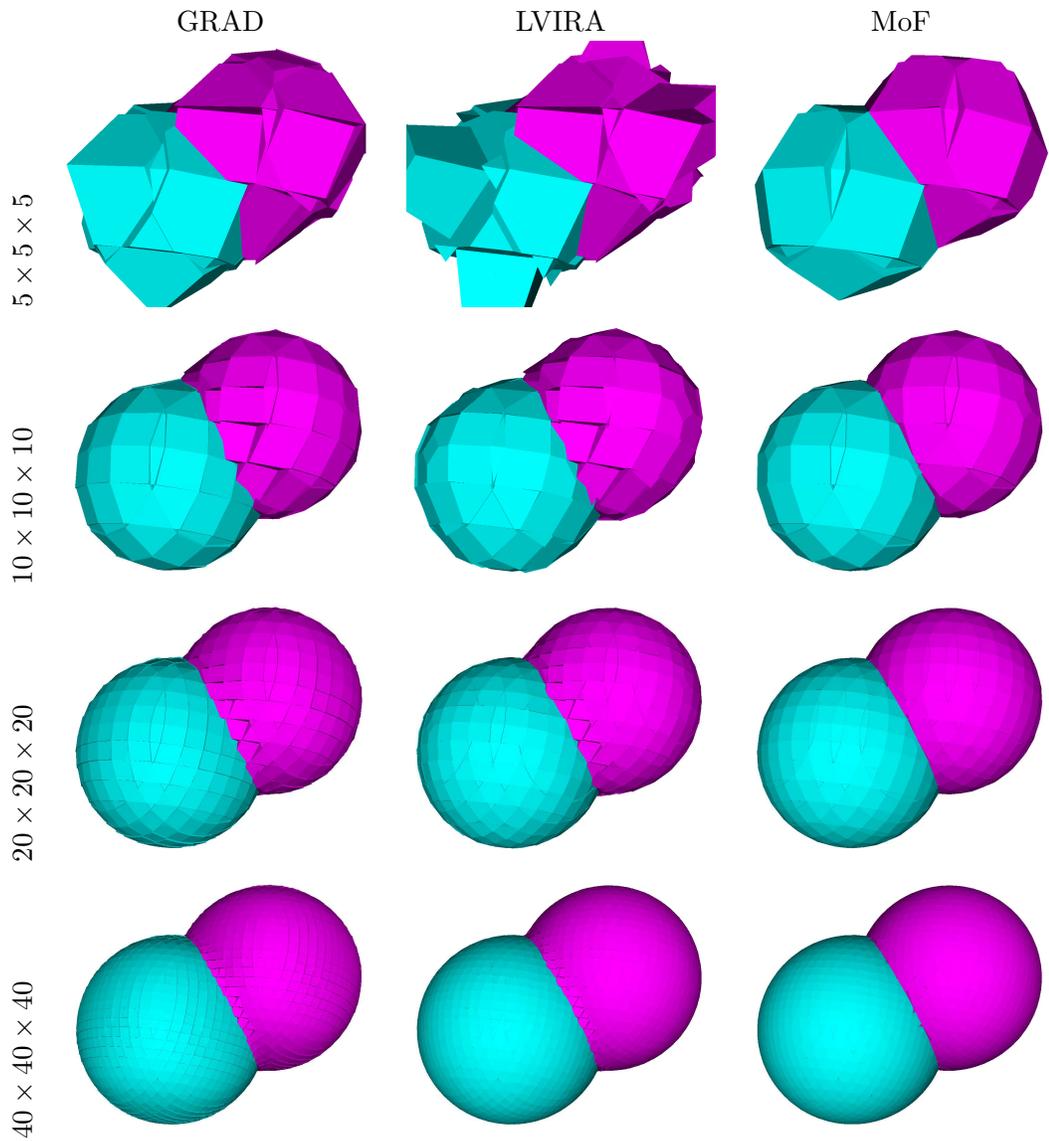


Figure 32: Mesh convergence study for multi-material case ( $n_{mat} = 3$ ). Global ordering of  $(\#1, \#2, \#3)$  are used for all reconstruction methods. For GRAD and LVIRA, interface normals are pre-computed by onion-skin strategy, then ND strategy is applied with given global ordering. Equispaced hexahedral meshes covering the domain of  $[0, 1]^3$  are used for interface reconstruction.

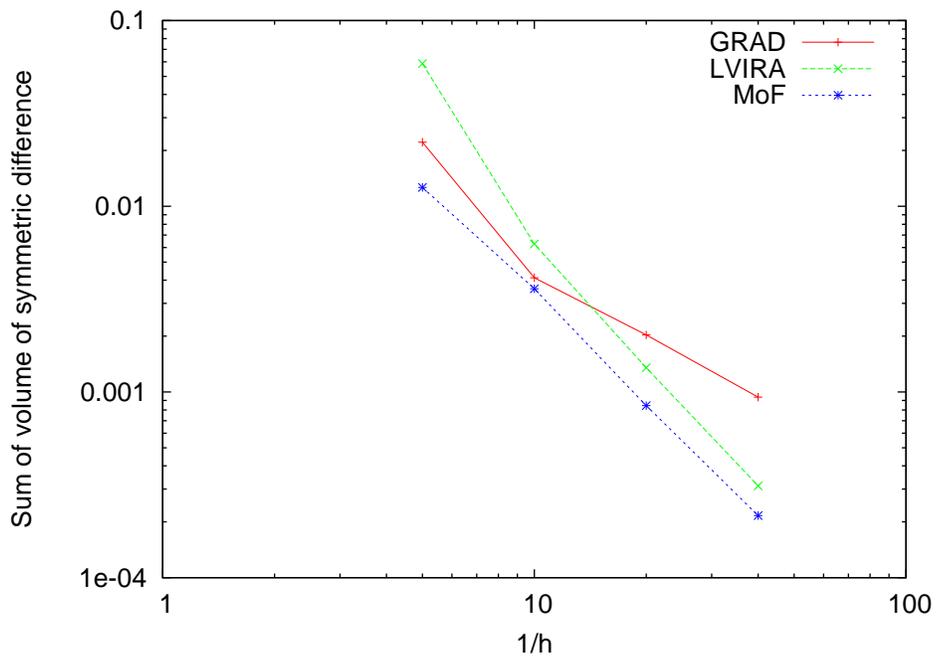


Figure 33: Rate of convergence for multi-material interface reconstruction. The convergence rates at the last refinement step are:  $p^{GRAD} = 1.169$ ,  $p^{LVIRA} = 2.034$ ,  $p^{MoF} = 1.966$

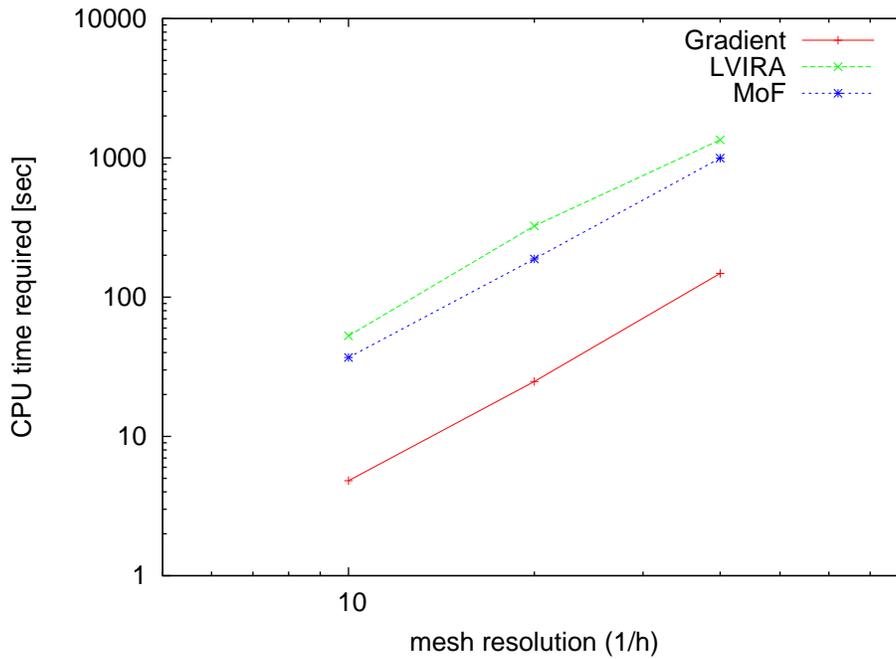


Figure 34: CPU time required with respect to mesh resolution for the reconstruction of three-material configuration displayed in Fig. 32. All three methods shows linear complexity ( $O(n)$  complexity, where  $n$  is number of mixed cells) with respect to the mesh resolution. At each step of successive refinement, the number of mixed cells quadruples and so the CPU time.

### 4.6.2 Examples of nested dissection

In this section, the effectiveness of automatic-local ordering strategy together with ND strategy (Sec. 4.1) is demonstrated with our primary reconstruction scheme, MoF. For GRAD and LVIRA, *onion-skin* ND strategy is employed as explained in Sec. 4.4. In onion skin ND, first interface normals are pre-computed by conventional onion skin model, and then ND strategy is applied with the pre-determined ordering given by MoF.

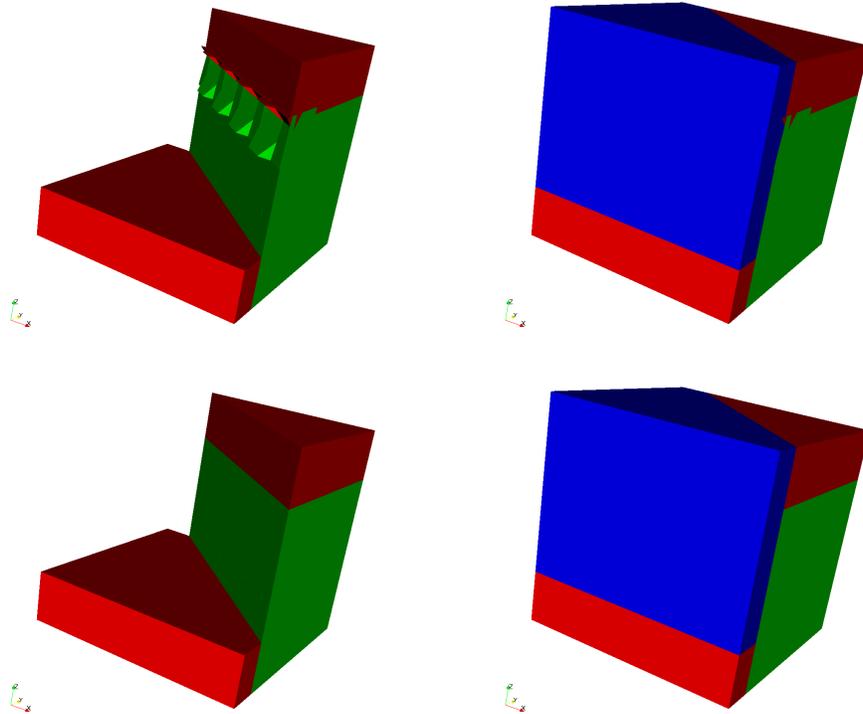


Figure 35: Global *vs.* local ordering for MoF reconstruction of interfaces with multiple triple-lines on  $5^3$  hexahedral mesh. Global ordering (top) cannot resolve all triple-lines but the local ordering (bottom) can resolve both due to its adaptation capability in ordering.

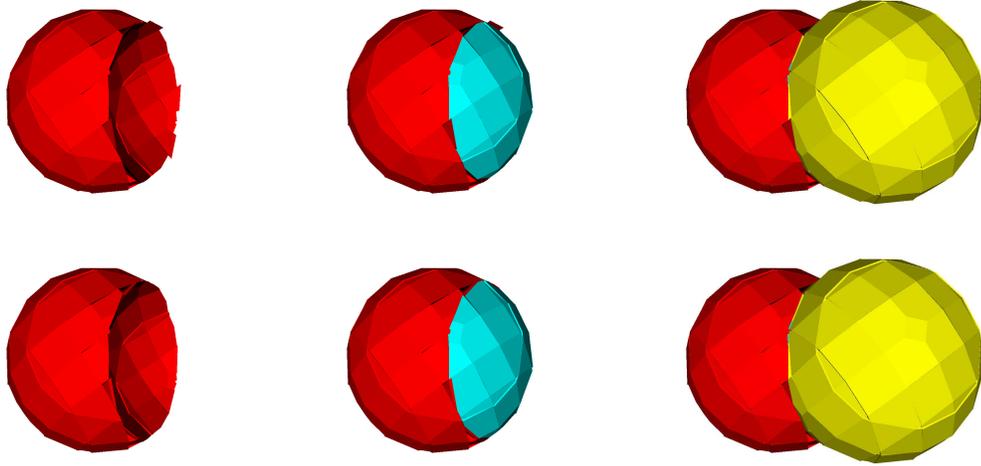


Figure 36: Global *vs.* local ordering for MoF reconstruction of two sphere configuration (as illustrated in Fig. 16) on  $10^3$  hexahedral mesh. Global ordering (top) cannot resolve clean intersecting edges (especially clear on left and middle columns of pictures), but the local ordering (bottom) resolves the clean edges with adaptive ordering from cell to cell. Each material (except the background) is displayed accumulatively from the left column.

**Local *vs.* Global ordering** The effectiveness of automatic local ordering can be demonstrated by MoF interface reconstruction with multiple triple lines as shown in Fig. 35. Carefully selected global ordering can recover a single triple line, but not all. On the other hand, local ordering, which automatically adapts ordering from cell to cell, can resolve both of the triple lines.

The advantage of automatic local ordering is further emphasized with Fig. 36. The global ordering is carefully chosen so that it has minimal variance from the automatic local ordering by MoF, hence it can be considered as a *ad hoc* optimal global ordering. Along the edges of intersection between two spheres, the global ordering shows rough reconstruction of red material. On the contrary, the automatic local ordering shows clean edge reconstruction, which confirms the superiority of automatic local ordering.

**Comparison of MoF with LVIRA and GRAD** In contrast to the straight forward extension of MoF to multi-material automatic-local order-

ing, LVIRA and GRAD has inherent restriction in multi-material generalization due to its non-locality. Hence, for the comparison purpose of the methods in multi-material case, the local orders obtained by MoF are utilized for the *onion-skin* ND interface reconstruction of LVIRA and GRAD.

In Fig. 37, the three methods are compared with double triple line example by using the local ordering determined by MoF. With automatic-local ordering, MoF recovers both of the triple lines, but LVIRA and GRAD failed to recover either one.

Fig. 38 emphasizes the superiority of MoF over other methods in multi-material case. MoF recovers the two intersecting spheres accurately, but both LVIRA and GRAD shows poor recovery of two sphere especially along the edge of intersection. The above two example strengthens the effectiveness of automatic-local ordering in MoF for multi-material interface reconstruction.

These results also demonstrate the weakness of onion-skin model for computation of normals in GRAD and LVIRA.

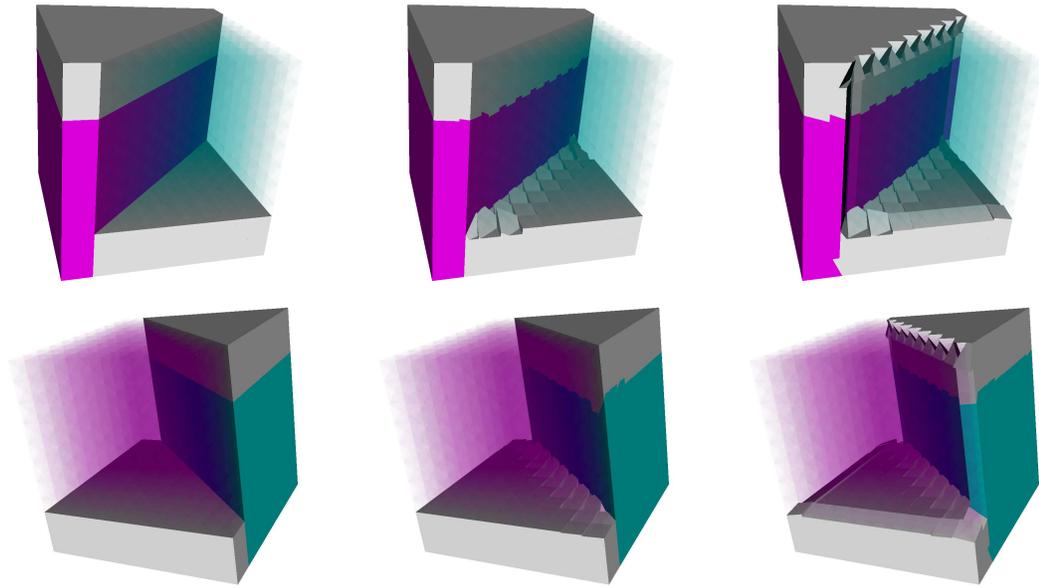


Figure 37: MoF, LVIRA, and GRAD (from the left) interface reconstruction of double triple lines with pre-determined local ordering by MoF. Double triple lines are reconstructed correctly by MoF with local ordering. However, *onion-skin* reconstructions of GRAD and LVIRA failed to recover the double triple lines even with the same local ordering as MoF. For visualization purpose, top row is inverted up-side-down at bottom with alternating transparent region.  $10^3$  hexahedral mesh is used.

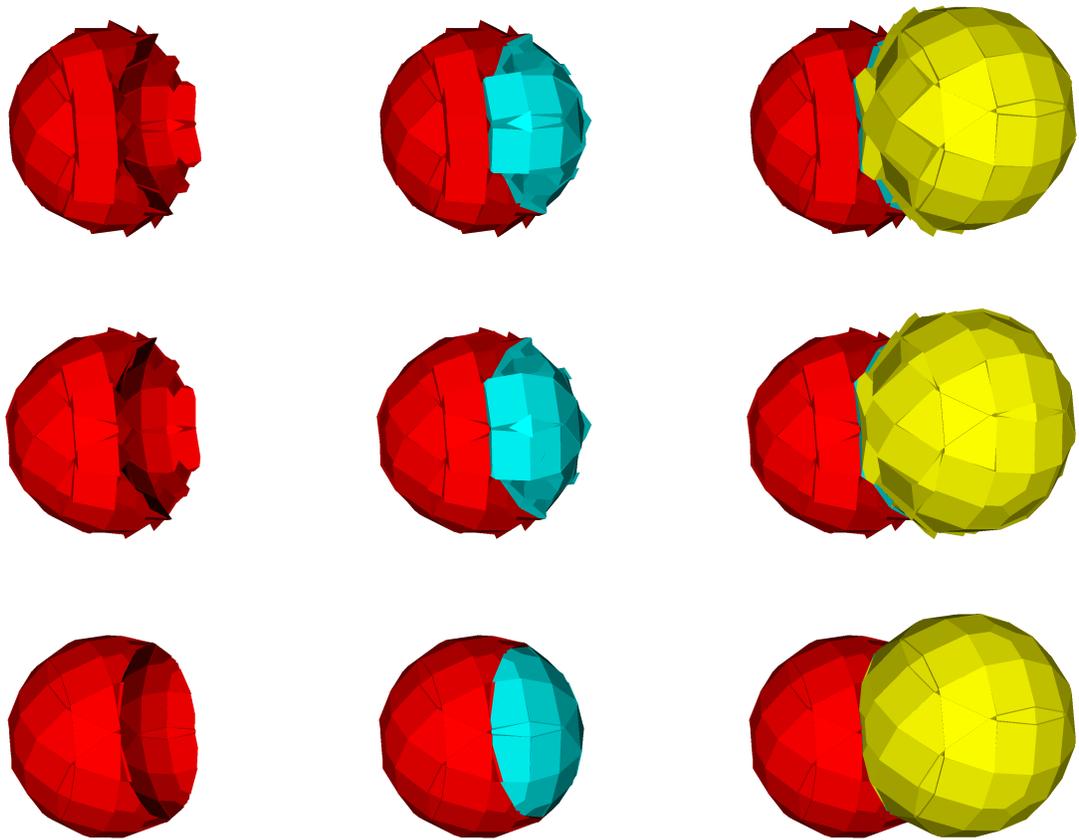


Figure 38: GRAD, LVIRA, and MoF (from the top) interface reconstruction of two sphere example with pre-determined local ordering by MoF. MoF shows best recovery of the two intersecting spheres with automatic-local ordering, but the *onion-skin* normals used in GRAD and LVIRA reconstruction methods poorly recover the intersecting spheres even with the same local ordering as MoF.  $10^3$  hexahedral mesh is used.

### 4.6.3 Examples of group nested dissection

The effectiveness of grouping in multi-material interface reconstruction is illustrated in Fig. 39. The computational domain containing four material is discretized into  $3^3$  hexahedral cells. In the cells contains all four materials, the nested dissection barely recovers only a single triple line, but grouping shows perfect recovery of both triple lines. In this particular interface configuration, a mesh refinement will not produce cells with two triple lines in this case. This grouping strategy, however, brings more capability even on coarse meshes.

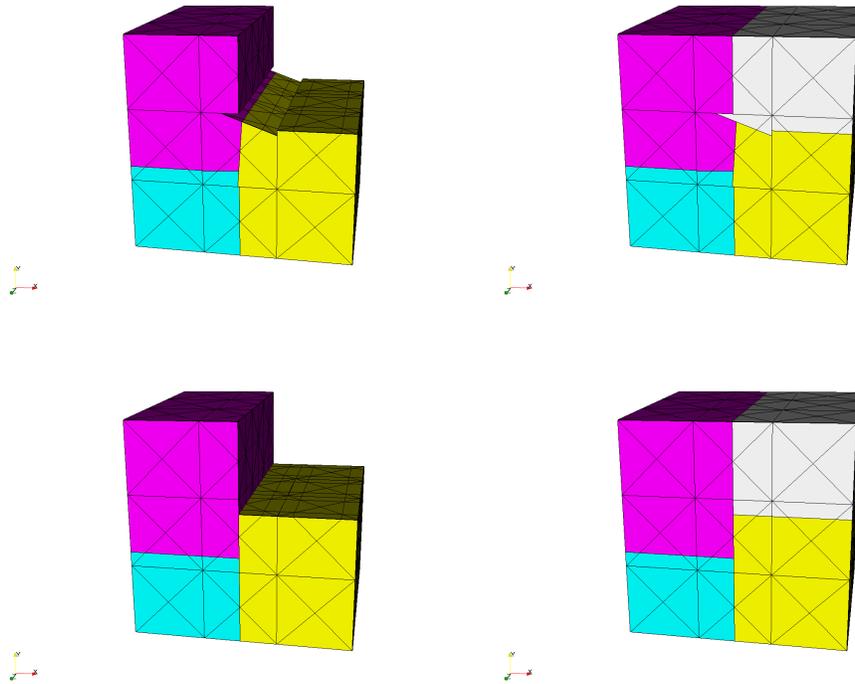


Figure 39: ND *vs.* GND. Nested dissection (top) cannot recover dual triple-lines at the central cell, but the group-nested dissection (bottom) recovers the triple-lines correctly.  $3^3$  hexahedral mesh is used.

The last example of GND is displayed in Fig. 40. Grouping is applied to the cells containing quadruple lines (+ junction). The cell with quadruple line inside contains four materials. In the case of quadruple lines, mesh re-

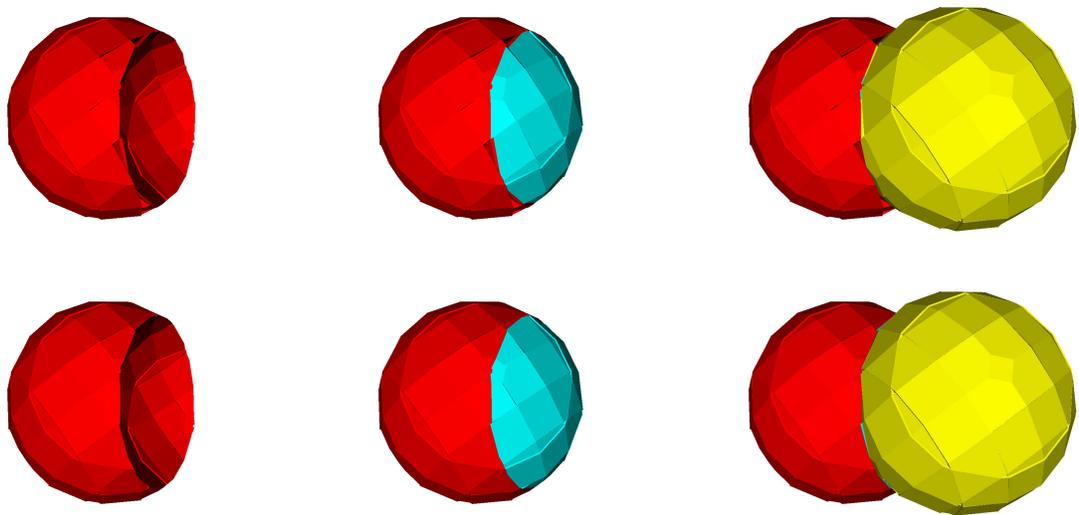


Figure 40: ND *vs.* GND with two sphere example. Nested dissection (top) shows wiggles around cells containing four materials (cells including quadruple lines), but the group nested dissection recovers spherical surface without wiggles.  $10^3$  hexahedral mesh is used.

finement cannot be a remedy because, no matter how many mesh refinement made, there will be always cells with quadruple lines. The effectiveness of grouping is clear in Fig 40. Nested dissection reveals discontinuity on the spherical surface near the quadruple lines, but the grouping removes such discontinuity, which emphasize the advantage of GND.

## 5 Conclusion and Future work

We have described multi-material (more than two materials) interface reconstruction methods for 3D mesh of generalized polyhedrons with non planar faces. All methods subdivide mixed cell into set of pure non-overlapping convex sub-cells each containing just one material. This is achieved by using nested dissection and group nested dissection schemes. It is shown that group nested dissection method allows to correctly reconstruct very complicated multi-material configurations. We have described three methods. First two methods represent extension of standard PLIC methods into 3D and use information only about volume fractions. GRAD method is first-order accurate and based on discrete gradient of volume fraction as estimate to normal to interface. LVIRA method is planarity preserving (second-order accurate) and is extension to 3D of least squares volume of fluid interface reconstruction algorithm. For these two methods there is no general strategy for material ordering as well as for computation of normals in multi-material case. MoF method is second-order accurate (planarity-preserving) This method uses information not only about volume fraction but also about position of the centroids for each material from the cell where reconstruction is performed, no information from neighboring cells is needed. In MoF method the objective is to choose normals to interfaces in such a way that distance between reference and centroids of reconstructed pure sub-cells is minimized, when exactly matching reference volume fractions. MoF method provides automatic ordering of the materials in process of interface reconstruction. Optimal local ordering is based on comparing positions of the reference and actual (centroids of pure sub-cells) positions of the centroids.

Performance of the methods is demonstrated on numerical examples that involve different types of meshes, smooth and non-smooth material shapes. Convergence study is performed for case of two materials as well as for multi-material case. Relative computational cost of describe methods is also presented.

Numerical examples show that MoF is most accurate method for all levels of the mesh resolution. Our opinion that ability of MoF method

to automatically choose optimal local ordering of the materials, locality of the method (which allows efficient parallel implementation) as well as its accuracy makes MoF method of choice.

In future we will describe how incorporate moment of fluid method into ALE framework. That is how to obtain information about reference volume fractions and reference centroids and how to to perform multi-material remap including remapping of the centroid positions.

**Acknowledgment** The authors like to thank V. Dyadechko, R. Garimella, M. Kucharik, S. Schofield, M. Berndt, K. Joy, D. Benson, D. Kothe, B. Rider, B. Swartz, D. Bailey, M. Francois, J. Mosso, T. Aslam, A. Robinson, D. Hardin, A. Barlow for helpful discussion and valuable comments. This work was supported by the Advanced Simulation and Computing (ASC) program at the Los Alamos National Laboratory.

## A Geometric algorithms for interface calculation

In this section, the geometric algorithms for interface reconstruction with GPM are presented. The algorithms for intersection of convex polyhedron with half-space is described. Regular case (no vertices on cutting plane) is first presented, and then degenerate (vertices on cutting plane) cases are discussed.

### A.1 Intersection of convex polyhedron with half-space

Convex polyhedron intersection with a half-space is the base operation for interface reconstruction in 3D. First the algorithm of intersection is presented for regular case (no vertices on cutting plane), and later issues and strategies for degenerate cases (vertices on cutting plane) will be addressed.

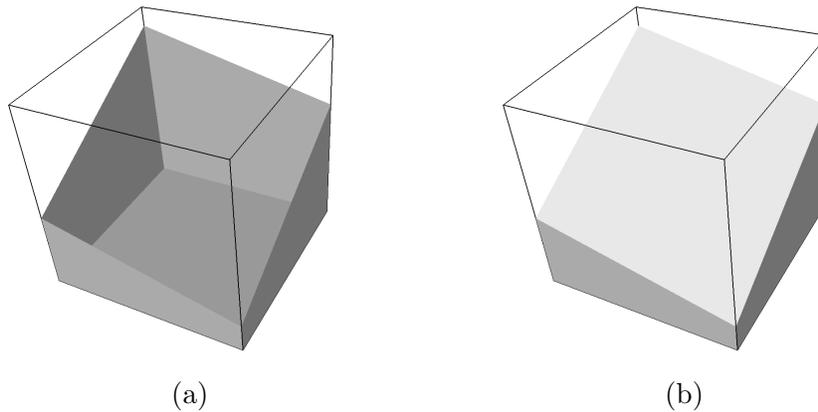


Figure 41: Convex polyhedron intersection by clipping and capping. (a) shows an open hexahedron by clipping the hexahedron with a given plane, and (b) illustrates the closed polyhedron by clipping as well as capping.



hedron be a closed polyhedron having all edges two neighbors, the slice face is identified by capping stage. In contrast to the clipping stage, the capping stage needs a proper orders of face-visits for slice face construction. The slice face is constructed by continuation of the slice curve as delineated in Fig. 42. The slice curve can get started with any given mixed face. The curve is continued by looping the adjacent mixed faces until it returns back to the initial mixed face and completing closed slice curve.

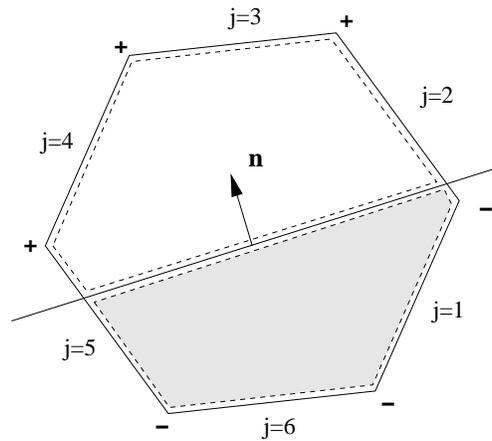


Figure 43: Convex polygon intersection with a plane of interface normal ( $\mathbf{n}$ ) in 3D. Polygon intersection routine returns two sub-polygons indicated by dashed lines: first sub-polygon which is below to the cutting plane (gray part) and second sub-polygon above (void part). New vertices are generated by intersection of the plane and edges with different signs ( $j = 2, 5$ ).

The convex polyhedron intersection algorithm incorporating both clipping and capping is illustrated in Algorithm 1. The inputs of the polyhedron intersection routine is initial *polyhedron* and cutting *plane*, and the outputs are two closed sub-polyhedra; *phed1* below the given cutting plane and *phed2* above the plane. Inside of the loop of mixed faces, polygon intersection is performed. Intersection of convex polygon with a plane in 3D is illustrated in Fig. 43. In the convex polygon intersection subroutine as described in Algorithm 2, like polyhedron intersection routine, the inputs are a *polygon* (a face of *polyhedron*) and cutting *plane*, and the outputs are two closed polygons; *pgon1* which is below the plane and *pgon2* which is above the plane. Polygon intersection algorithm is similar to that of polyhedron intersection

as presented in Algorithm 1. The main difference of polygon intersection is edge-wise loop is carried out instead of face-wise loop in polyhedron intersection.

```

Input: polyhedron, plane
Output: phed1, phed2
foreach face of polyhedron do
  if unvisited face then
    if face below to plane then           /* pure face */
      add this face to phed1 ;
      mark this face visited ;
    else if face above to plane then     /* pure face */
      add this face to phed2 ;
      mark this face visited ;
    else if face gets intersection then /* mixed face */
      face_start  $\leftarrow$  this face ;
      repeat
        perform polygon/plane intersection ;
        add face below to phed1 ;
        add face above to phed2 ;
        mark this face visited ;
        face_next  $\leftarrow$  next face ;
      until face_start = face_next ;
    end
  end
end

```

**Algorithm 1: Convex polyhedron intersection**

**Degenerate cases.** Cutting plane does not always intersect edges (break the edge into two parts), and either or both of the vertices can be exactly on the cutting plane. For example, the vertices of polyhedron can be exactly (or within some tolerance) on the given plane. This results in degenerate cases, as delineated in Fig. 44. Degenerate cases requires two additional considerations. First, in polygon intersection routine, additional vertex may not be generated by intersection (of plane and edge in 3D), instead an ex-

```

Input: polygon, plane
Output: pgon1, pgon2
foreach edge of polygon do
  | if edge below to plane then      /* both vertices (-) */
  |   | add this edge to pgon1 ;
  | else if edge above to plane then /* both vertices (+) */
  |   | add this edge to pgon2 ;
  | else if edge gets intersection then /* vert's sign mixed */
  |   | perform segment/plane intersection ;
  |   | add edge below to pgon1 ;
  |   | add edge above to pgon2 ;
  | end
end

```

**Algorithm 2: Convex polygon intersection**

isting vertex is used for it. Second, for the continuation of the slice curve as delineated in Fig. 45, next adjacent face should be found carefully because not only the mixed faces but also the pure faces (if an edge is on the cutting plane) can be the candidate for the next face.

**Test cases.** The convex polyhedron intersection algorithm is applied for more general cases in Fig. 46. Two polyhedra intersected by the present algorithms are displayed. First the intersection of truncated icosahedron (*a.k.a* soccer ball geometry with 12 pentagons and 20 hexagons) is presented, and next the polyhedral representation of bunny (725,000 triangular surface mesh) is intersected. The bunny geometry is not an example of convex polyhedron, but as long as the slice face is *simply connected* and the polyhedron is convex-faced (triangulated surface here) the current algorithm can be applied.

**Volume and centroid computation** For each polyhedron intersection, volume and centroids of the intersected subcell have to be computed for measuring the error of interface reconstructed. For this purpose, fast and accurate computation of moment data of general polyhedron is indispensable, and our implementation is based on [38]. The algorithm is based on multi-step reduction of the volume integral to successively lower dimensions by using Divergence and Green's theorems. This method is also applied to the integraion of a linear function over an arbitrary polyhedron [18].

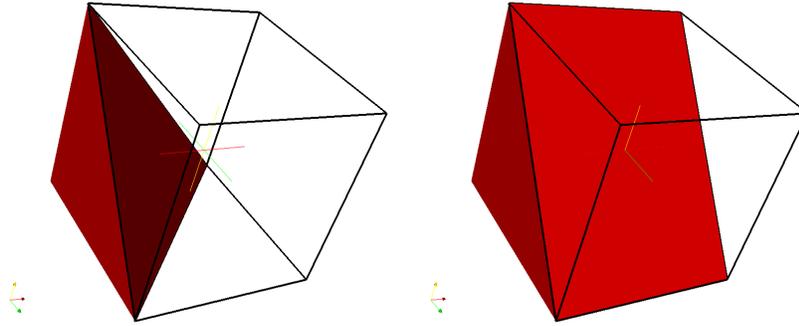


Figure 44: Degenerate cases in polyhedron intersection: vertices on the intersecting plane (left), and vertices as well as edges on the intersection plane (right).

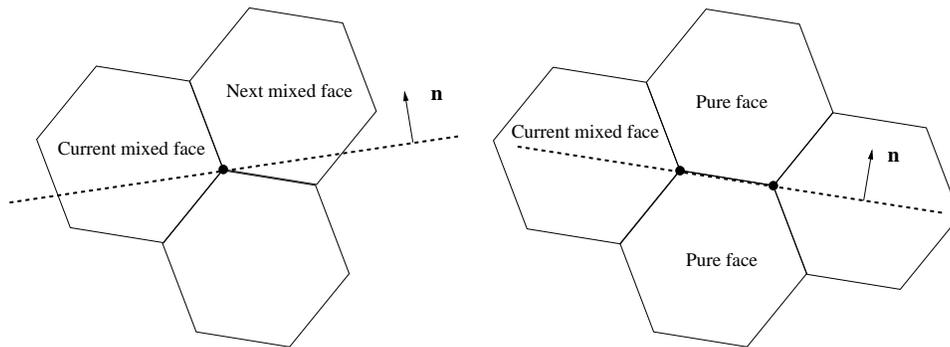


Figure 45: Degenerate cases in polygon intersection: one vertex on the intersecting plane (left), and two vertices (edge) on the intersection plane (right). Cutting plane is delineated with dashed line, and vertices on the plane is marked with  $\bullet$ .

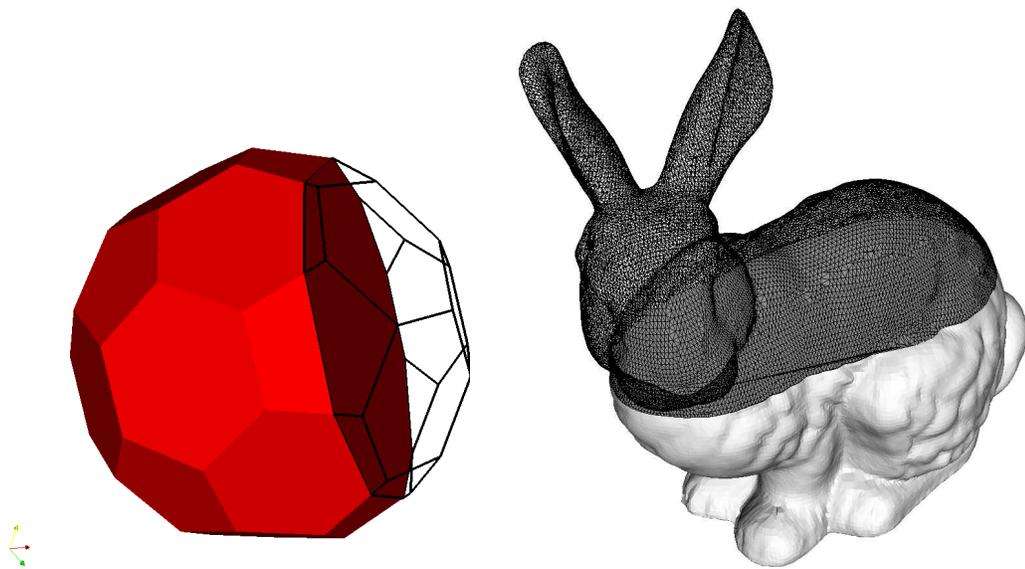


Figure 46: Intersection of complex polyhedra. Left shows 32 faced truncated icosahedron, and right shows 725,000 faced bunny mesh. Both surface mesh represents a polyhedron.

## B Volume matching intersection of generalized polyhedron

In this section, the target volume fraction matching interface calculation, with *given normal*, methods are presented.

### B.1 Volume preserving interface calculation

The primary mechanism of volume preserving interface reconstruction requires cutting appropriate volume fraction of the cell, as expressed in Eq. (8), and repeated as follows

$$\mathcal{V}(d) = \mathcal{V}^{ref}$$

where  $\mathcal{V}(d)$  is the volume of subcell below to the cutting plane (with given normal) defined by  $d$ , distance of cutting plane as expressed in Hessian normal form in Eq. (3), and  $\mathcal{V}^{ref}$  is reference (target) volume. The above equation can also be expressed as follows

$$f(d) = f^{ref}$$

where  $f(d) = \mathcal{V}(d)/\mathcal{V}_{cell}$  is volume fraction defined by  $d$  and  $f_{ref} = \mathcal{V}^{ref}/\mathcal{V}_{cell}$  is reference (target) volume fraction, which are both normalized by cell volume  $\mathcal{V}_{cell}$ . Since the normal (orientation) of cutting plane is given, the volume of intersection is purely function of distance,  $d \in [d_{min}, d_{max}]$ . For example,  $f(d_{min}) = 0$  and  $f(d_{max}) = 1$ .

Several approaches are proposed, but they are mainly described in 2D. These methods, *e.g.* analytical method [14] and semi-iterative method [48], require two pre-processing: first vertex-wise volume fraction evaluation ( $O(n)$  volume fraction evaluation) and then another vertex-wise volume fraction sorting ( $O(n \log n)$  operations in sorting), where  $n$  is number of vertices.

The analytical approach can be extended for tetrahedral cell in 3D [63]. For cells with small number of vertices, such as triangles in 2D and tetrahedra in 3D, this pre-processing and analytical approach could save CPU time. As the cells contains more vertices, typical for 3D polyhedral cells, these pre-processing demand considerable amount of CPU time as well as extra memory space besides the implementation efforts.

In order to cut target volume fraction accurately as well as efficiently two fully iterative schemes are employed, namely secant method and bisection method. The algorithm for the iterative methods is described in Algorithm 3.

**Input:**  $f^{ref}$ ,  $\mathbf{n}$ ,  $d_{min}$ ,  $d_{max}$   
**Output:**  $d$   
 $d_1 = d_{min};$   
 $d_2 = d_{max};$   
 $f_1 = 0;$   
 $f_2 = 1;$   
**repeat**  
    **if** *Secant method used* **then**  
        |  $secant = (f_2 - f_1)/(d_2 - d_1);$   
    **end**  
    update  $d$  by Secant or Bisection method;  
    intersect cell with defined interface  $(\mathbf{n}, d);$   
    compute  $f(d);$   
     $\Delta f = |f^{ref} - f(d)|;$   
    update  $d_1, d_2, f_1, f_2;$   
**until**  $(\Delta f < tol)$ ;

**Algorithm 3: Iterative volume fraction matching interface computation**

These iterative schemes have two distinctive advantages:

1. no pre-processing: vertex-wise volume fraction evaluation or sorting
2. fixed number of iteration regardless of number of vertices

First, no vertex-wise volume fraction evaluation or sorting is needed. For the start of the iteration, only the minimum and maximum distances with respect to the given interface normal are required. This is because of the monotonically increasing behavior (actually  $C^1$  if the cell is convex and  $C^0$  if not) of volume fraction with respect to the distance. Second, both iterative schemes are converging in almost fixed number of iterations regardless of cell size. In bisection method, with unit interval of distance  $[0,1]$  the number of iteration required to achieve distance error tolerance of  $tol = 10^{-10}$  is

$$\log_2 \frac{1}{10^{-10}} = 33.2193$$

regardless of function behavior [24]. Due to monotonicity of the function, the volume fraction error tolerance of  $tol = 10^{-10}$  is also achieved with approximately same number of bisection iterations as shown in Fig. 47.

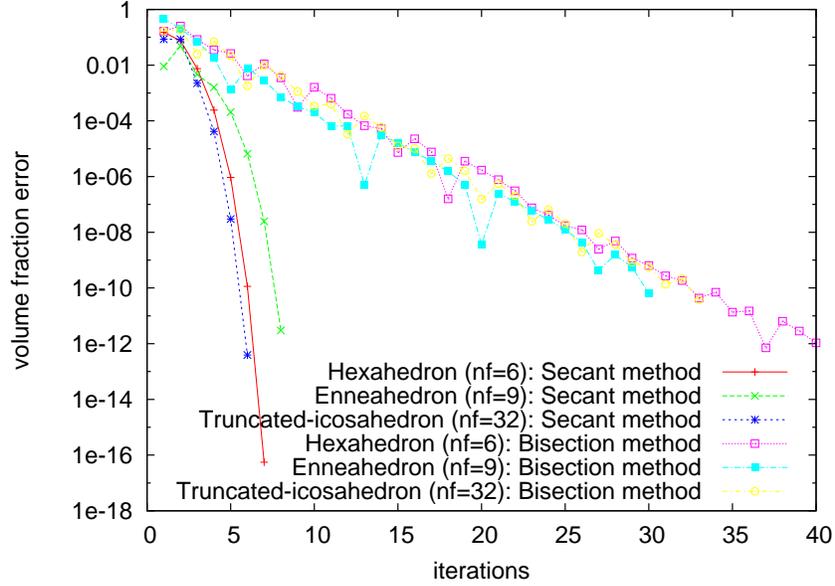


Figure 47: Volume fraction convergence of secant and bisection methods with three polyhedral cells shown in Fig. 6.

Fig. 47 shows the volume fraction convergence history of the two iterative methods applied to three polyhedral cells shown in Fig. 6. First, the secant method shows super linear convergence in volume fraction error. Less than 10 iterations are required to achieve the volume fraction error  $< 10^{-10}$ . Bisection method shows linear convergence, but it guarantees that only fixed number of iteration is required regardless of the number of vertices,  $n$ , for the cell.

The efficiency of iterative methods are further demonstrated with large size spherical cells as displayed in Fig. 48. Four levels of successively refined spherical surface meshes are used as a single polyhedral cell representation. In Fig. 49, using the four levels of spherical cells, the number of secant iteration to achieve  $err(f) < 1.e-10$  is measured with target volume fraction between  $[0,1]$ . The number of iteration required is irrespective to the size of polyhedral cell, and the target volume fraction is achieved almost less than 10 iterations.

For very small volume fractions ( $f \rightarrow 0^+$ ) or almost full cell (volume fraction close to one) ( $f \rightarrow 1^-$ ), the secant method requires many iterations and may fail to converge. This is because the derivative is almost zero as

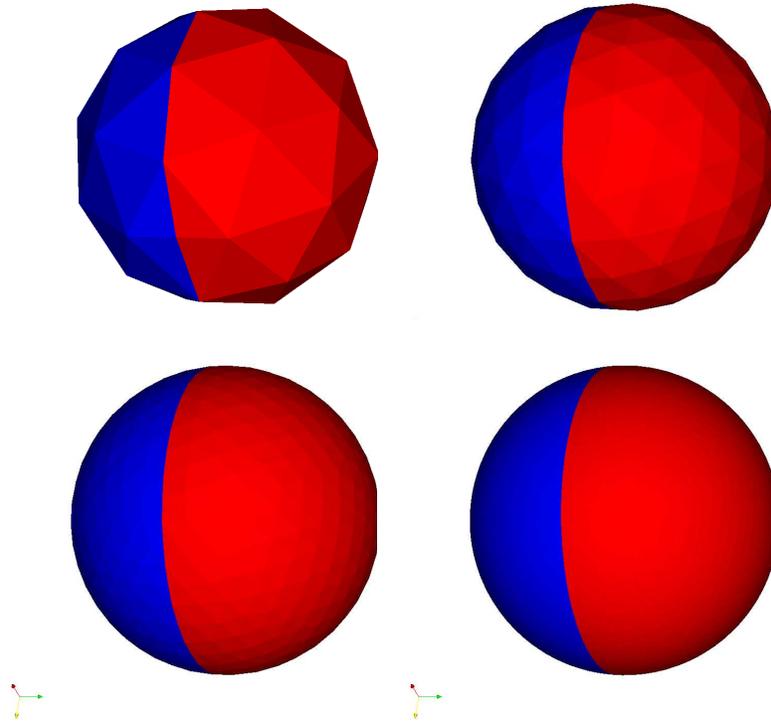


Figure 48: Spherical cell containing two materials (blue and red) with successive refinement. From the coarsest mesh to the finest, the numbers of faces are 80, 320, 1280, and 5120.

shown in Fig. 50 at the two limiting cases of volume fraction.

This behavior of volume fraction for "spherical" cell is displayed in Fig. 50. Whenever secant method fails to converge (*i.e.* either too many iteration required or an iterate goes away from the given distance interval  $[d_{min}, d_{max}]$ ), bisection method, which is always converging, is used.

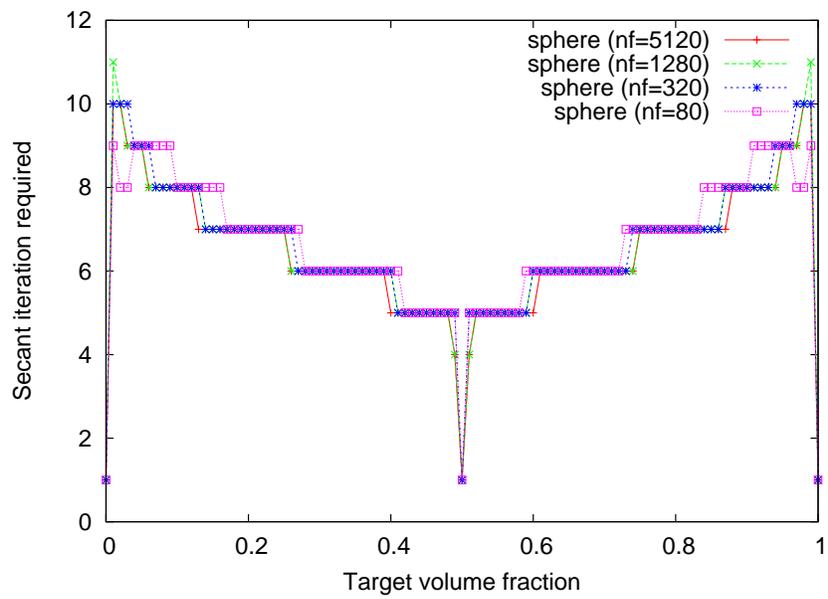


Figure 49: Number of secant iterations required for appropriate volume fraction cutting measured with four levels of spherical cells displayed in Fig. 48

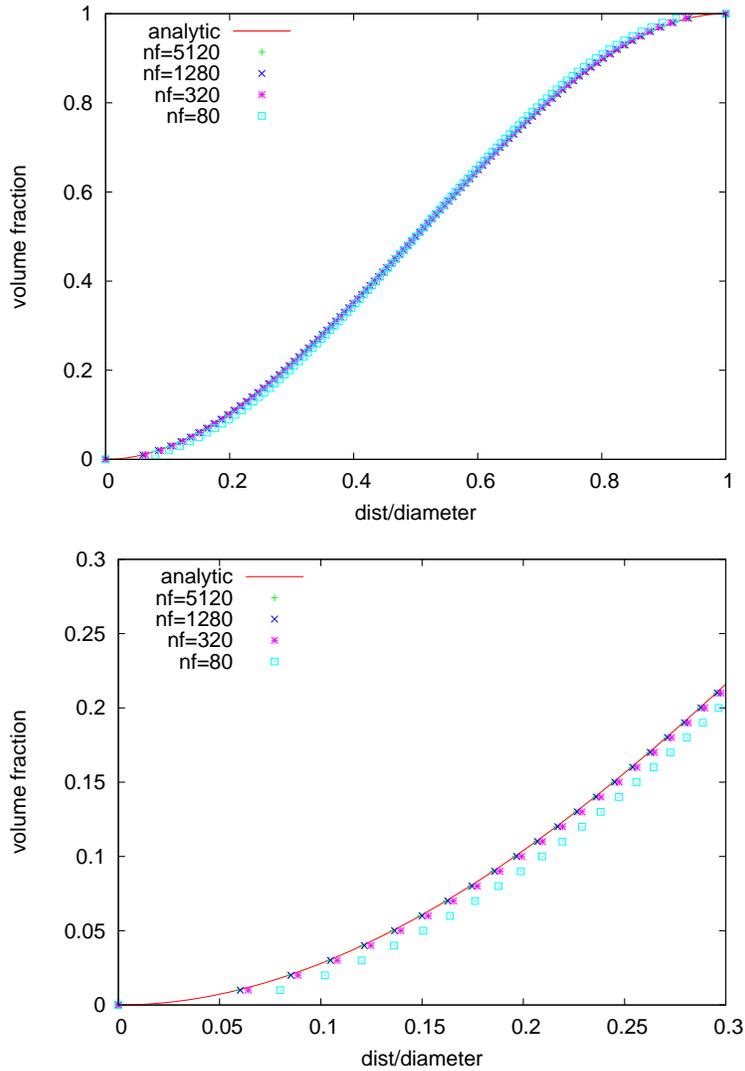


Figure 50: Spherical cell volume fraction behavior with respect to the distance. Top figure shows volume fraction with entire interval of distance, and bottom shows close-up view near the minimum distance. As mesh refines, volume fraction obtained by polyhedral cell converges to the analytic solution.

## C Numerical optimization

The second-order accurate interface reconstruction methods, LVIRA and MoF, require additional optimization process minimizing the error functionals. LVIRA error functional is expressed in Eq. (9), and MoF in Eq. (10).

In both case optimization has to be performed with respect to the normal  $\mathbf{n}$  in the Eq. (3):

$$\mathbf{n} \cdot \mathbf{r} + d = 0.$$

In 3D normal  $\mathbf{n}$  is defined by two polar angles  $(\phi, \theta)$  as illustrated in Fig. 51.

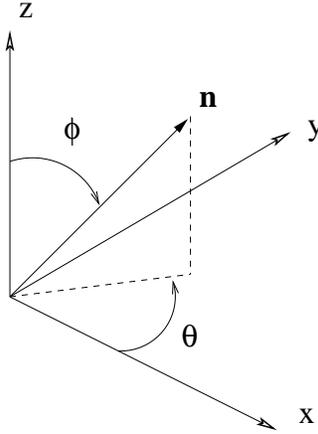


Figure 51: In 3D interface normal  $\mathbf{n}(\phi, \theta)$  is defined by two polar angles ( $\phi \in [0, \pi]$  and  $\theta \in [0, 2\pi]$ ).

By using the spherical interface configuration delineated in Fig. 52, typical behavior of the objective functions are displayed in Fig. 53. For the interface configuration in Fig. 52, sphere centered at  $(-0.1, -0.2, -0.3)$  with radius  $r = 1.3$  is used and equispaced  $3 \times 3 \times 3$  hexahedral mesh covering the domain of  $[0, 1] \times [0, 1] \times [0, 1]$  is used. For visualization purpose, a transparent fraction of the sphere and the cell centered at  $(0.5, 0.5, 0.5)$  are displayed.

The behavior of objective functions for the center cell shown in Fig. 52 are displayed in Fig. 53. General trends of these objective functions are

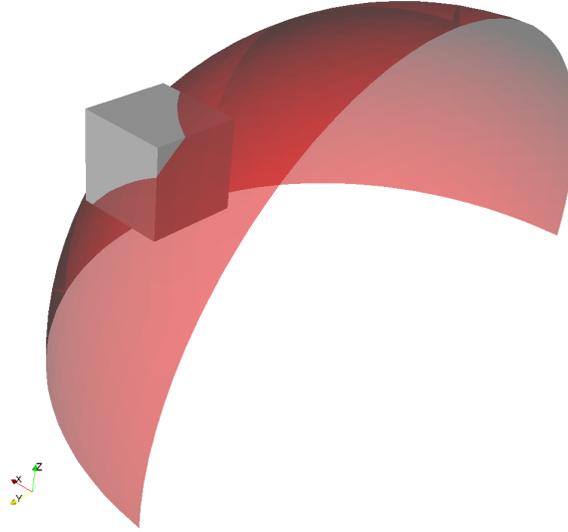


Figure 52: Interface configuration by a sphere, centered at  $(-0.1, -0.2, -0.3)$  with radius  $r = 1.3$ , and equispaced  $3^3$  hexahedral mesh covering the domain of  $[0, 1]^3$ . For visualization purpose, a fraction of transparent sphere and the central cell of the mesh are displayed.

similar for both LVIRA and MoF. However, the scale of absolute values of the functions are different. This is because LVIRA uses accumulated volume fraction difference from neighbors and MoF uses normalized distances between centroids as the objective function.

The current numerical optimization is *local* minimization, *i.e.* finding the closest local minimizer to given initial guess.

For the above multi-dimensional minimization, Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [27] is used. It is a quasi Newton method, approximating Hessian matrix with a set previous of gradients. The gradients of the objective function are computed by finite differences. For each search direction, a quadratic or cubic polynomial line search is performed for sufficient decrease in the error with the Armijo rule for step size control. Detailed discussion of the BFGS method can be found in [27]

The actual convergence path of the optimization routine is displayed in Fig. 54. Solution at each stage of iteration is indicated with  $\circ$  on the contour plots. For the initial guess of the optimization, the gradient of volume fraction computed as in GRAD is utilized for LVIRA.

$$\mathbf{n}_0(\phi_0, \theta_0)_{LVIRA} = -\mathbf{GRAD}(f). \quad (16)$$

For MoF, the unit vector along the given material centroid to the cell centroid is used as follows

$$\mathbf{n}_0(\phi_0, \theta_0)_{MoF} = \frac{\mathbf{x}_c^{cell} - \mathbf{x}_c^{ref}}{\|\mathbf{x}_c^{cell} - \mathbf{x}_c^{ref}\|} \quad (17)$$

where  $\mathbf{x}_c^{cell}$  is cell centroid and  $\mathbf{x}_c^{ref}$  is the reference material centroid. Once the gradient of the objective function becomes less than a given tolerance, it is considered that a local minimum is found and the optimization process terminates.

The convergence history of the LVIRA and MoF are displayed in Fig. 55. Both LVIRA and MoF show super-linear convergence rate, and  $\|\nabla(Error)\| < 10^{-6}$  are achieved with 10 iterations.

The final reconstructed interface for the mixed cell configuration shown in Fig. 52 is displayed in Fig. 56. Fraction of original spherical interface is overlapped with transparency. Depending on the reconstruction methods, the interface normal is different and this results different interface reconstruction as shown in the figure. The volumes of symmetric difference, as expressed in Eq. (11), at the cell are measured as follows: 6.1616e-04 (GRAD), 5.9999e-04 (LVIRA), 5.5676e-04 (MoF). Again this result strengthens that MoF gives the best accuracy.

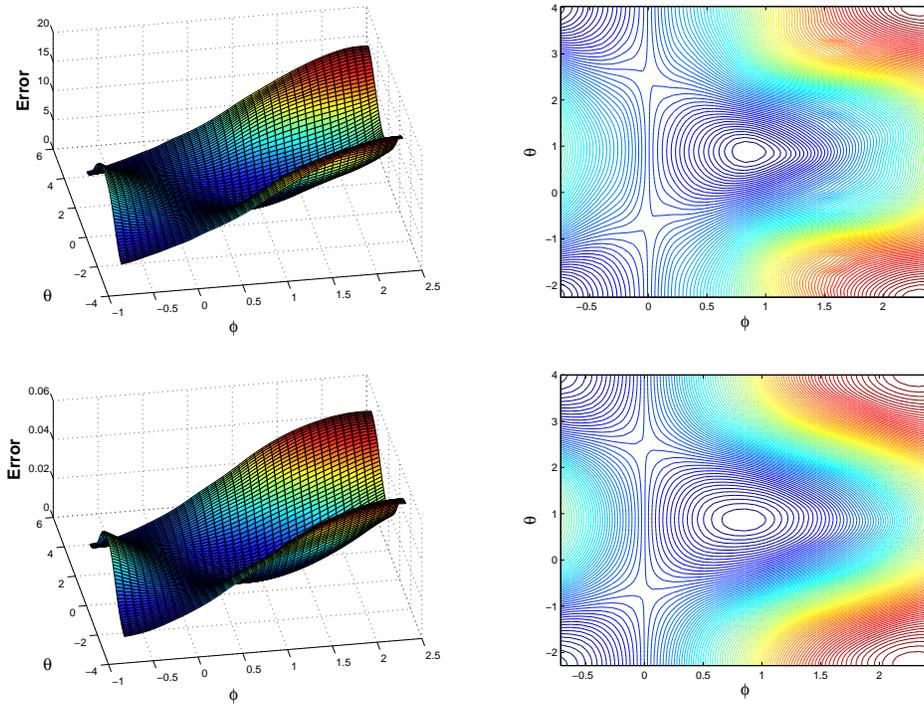


Figure 53: Behavior of objective function of a mixed cell: Top — LVIRA, and bottom — MoF. Left column is 3D view of the objective functions, and right is contours on the plane of two polar angles  $(\phi, \theta)$ . Global shapes of the functions are similar for both LVIRA and MoF. However, the scale of absolute values of the objective function are different, because LVIRA uses volume fraction difference and MoF uses normalized distances between centroids.

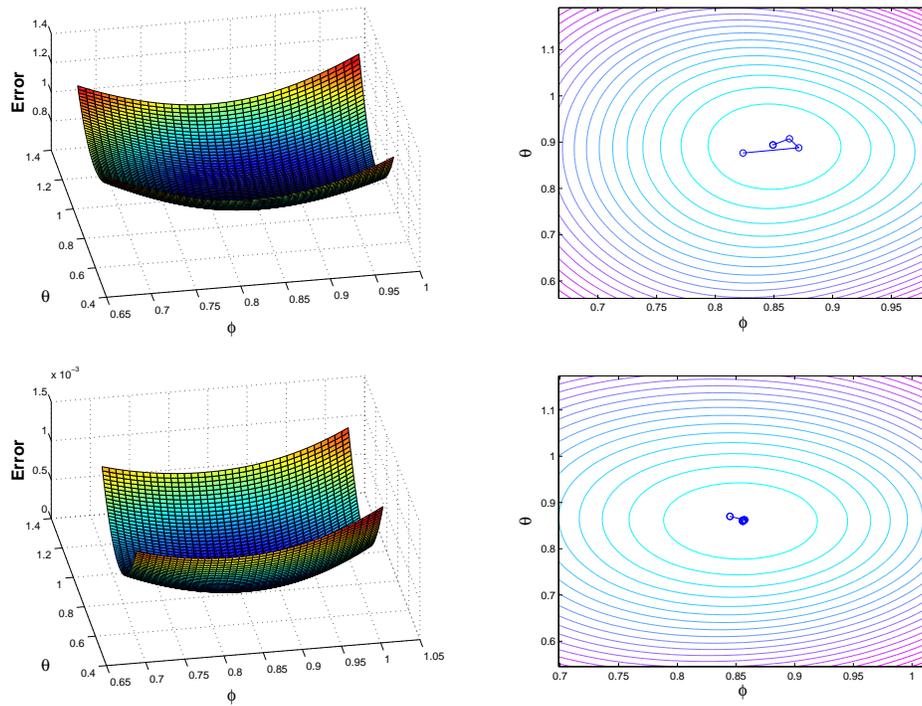


Figure 54: Convergence of minimization process: Top — LVIRA, and bottom — MoF. Left column is close-up view of the objective functions shown in Fig. 53, and right is the corresponding contours on the plane of two polar angles  $(\phi, \theta)$ . Each iteration of the minimization is indicated with  $\circ$  on the contour plots.

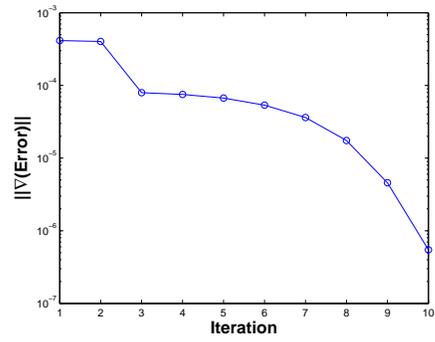
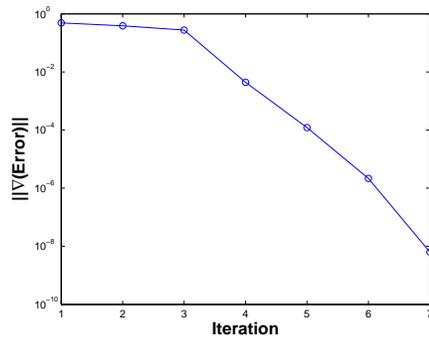


Figure 55: Convergence history of optimization. Left–LVIRA, Right–MoF. Both cases show super-linear convergence rate.

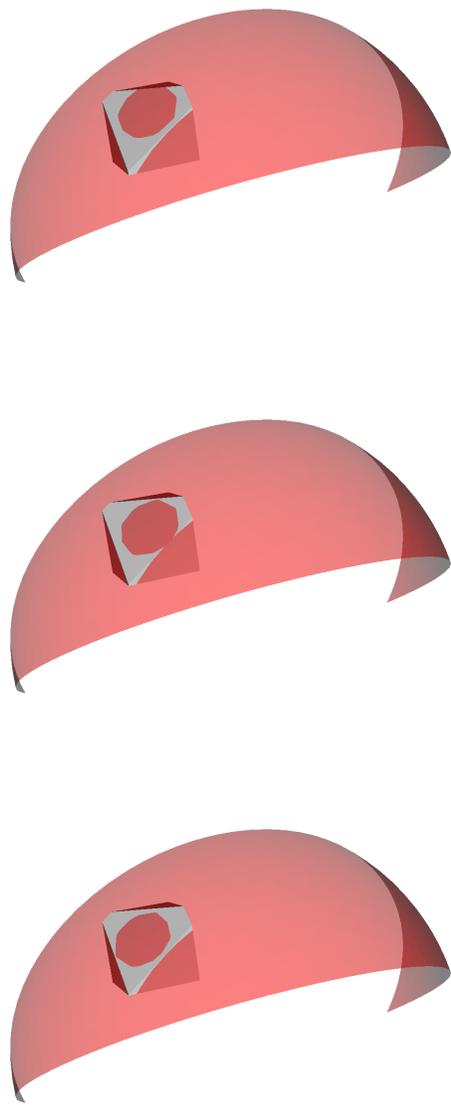


Figure 56: Interface reconstruction of the configuration shown Fig. 52. Top – GRAD, middle – LVIRA, and bottom – MoF.

## D Initialization of volume fraction and moment data

Real ALE simulation starts with initialization of all flow parameters. In particular fluid volumes (and centroids for some methods) are initialized in each computational cell from specified geometry. This task requires computing fluid volumes in each cell (mixed cell) containing interface(s). Exact interface information is then discarded in favor of the discrete volume (and centroids for some methods) data. Then, as it is described in introduction, special closure models are used to evolve fluid volumes forward in time and some special algorithms can be used to advance position of the centroids. Clearly this processes introduce some errors. Therefore only after initialization information about volume fractions and centroids can be considered "exact". Analysis of the errors due to time advancement of the volume fractions and centroids is far beyond of the scope of this paper.

In this paper, we only interested in accuracy of interface reconstruction and therefore, we will assume that volume fractions and centroids information which is used by interface reconstruction methods is "exact".

Computationally efficient strategy, for initialization of volume fraction and centroid data, consist of two steps:

1. Detect mixed cells,
2. Compute initialization data only for the mixed cells.

We assume that a valid geometric model for the object of interest is provided to us. This can be a CAD model or solid volume mesh or in simplest case body can be given "analytically", for example, as system of inequalities or Boolean operation between the given models. An example of mixed cell detection based on a simple spherical model (analytically described) is displayed in Fig. 57.

In this paper, for most complex geometries we assume that we are given a surface mesh, which can be readily obtained from any valid geometric model. Once the surface mesh of geometric model is provided, then it is scaled properly and immersed into the base mesh to be used for interface reconstruction.

Now, the mixed (interface) cells of the base meshes has to be detected. For this step, we first consider smooth shapes and then non-smooth shapes.

As an example of smooth shape, we use surface mesh of knot-shape obtained from [20], as displayed in Fig. 58. For smooth shapes like the knot,

interface cells can be accurately identified by checking if the cell vertices are in/out (we use ray crossing algorithm for this purpose due to its low computational cost compared with other algorithms, such as solid-angle algorithm [42]) of the shape even on relatively coarse mesh.

For non-smooth shapes, such as cow geometry with horns and ears as shown in Fig. 59, slender sharp corners may not be detected by simple algorithm unless the base mesh is extremely fine. For this case, additional process has to be introduced, namely *hidden mixed cell detection*. The suspicious cells (pure cells, as identified by first stage, attached to mixed cells) has to be examined with enriched interior points in search of hidden mixed cells. Any detection of a hidden mixed cell defines new suspicious cells around it. Once, a suspicious cell is identified as pure cell by enriched interior point check, then it can be marked accordingly to avoid double check by the hidden cell detection algorithm. The hidden cell detection continues until all suspicious cells are confirmed as pure (no more suspicious cell left). This process of hidden interface cell detection is displayed in Fig. 59.

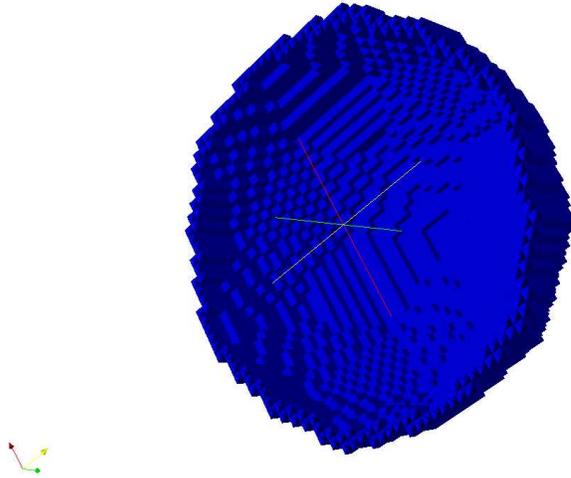


Figure 57: Detection of mixed cells for initialization of spherical interface reconstruction.  $40^3$  hexahedral mesh is used as the base mesh.

Once the mixed cells are identified, the volume fraction and moment data

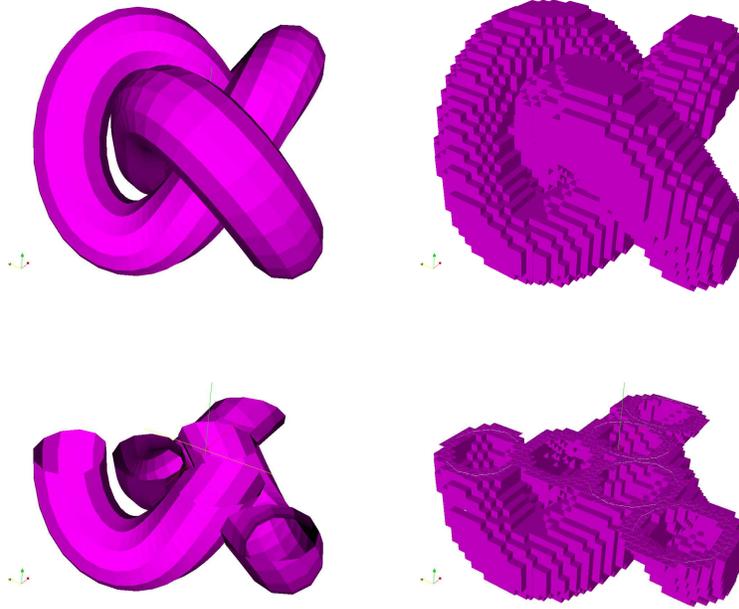


Figure 58: Detection of interface cell with an immersed surface mesh. Left column is original surface mesh and right column is detected interface cell of the base  $20^3$  hexahedral mesh. Top row is view of entire geometry, and bottom of clipped view revealing the inner hollow for pure cells.

have to be computed for these cells. Simple point in/out polyhedron test can be used also for this purpose as follows. One uniformly spaces points in a bounding box of a mixed cell, and perform in/out test of the points with respect to the given geometric model and cell under consideration. Since the number of interface cells compared to total cells are order of magnitude small (this will be more evident as mesh gets fine), the particle test is feasible for the present study. In particle test, volume fraction ( $f$ ) and centroids ( $\mathbf{x}_c$ ) can easily computed as follows

$$f = \frac{n_{obj}^p}{n_{cell}^p}, \quad \mathbf{x}_c = \frac{\sum_{j \in \mathcal{P}_{obj}} \mathbf{x}_j}{n_{obj}^p},$$

where  $n_{cell}^p$  is the number of particles inside of the cell,  $n_{obj}^p$  is the number of particles inside of the cell as well as the object (to reconstruct),  $\mathcal{P}_{obj}$  is the set of particles inside of both the cell and object and  $\mathbf{x}_j$  is the coordinates of  $j$ -th particle in set  $\mathcal{P}_{obj}$ . It should be noted that the distribution of particles

for initialization is equispaced.

This strategy can be easily extended to multi-material case. For example, a single surface mesh can be used initialization of up to  $2^1$  materials. Hence,  $n$  surface meshes can be used for initialization of up to  $2^n$  materials (provided that all surfaces intersect each other). For example, as delineated in Fig. 60, the two mutually intersecting geometric objects (here circle and triangle represents surface meshes of two different objects, *i.e.*  $n = 2$ ) can describe up to  $2^n = 4$  unique material regions. The number of material ID can also be assigned in lexicographical orders as shown in Table 2.

model #1 (circle)	model #2 (triangle)	material ID
0	0	#1
0	1	#2
1	0	#3
1	1	#4

Table 2: Assignment of material ID based on in/out decision with given geometric models. 0 represents out of the model and 1 represents inside of the model.

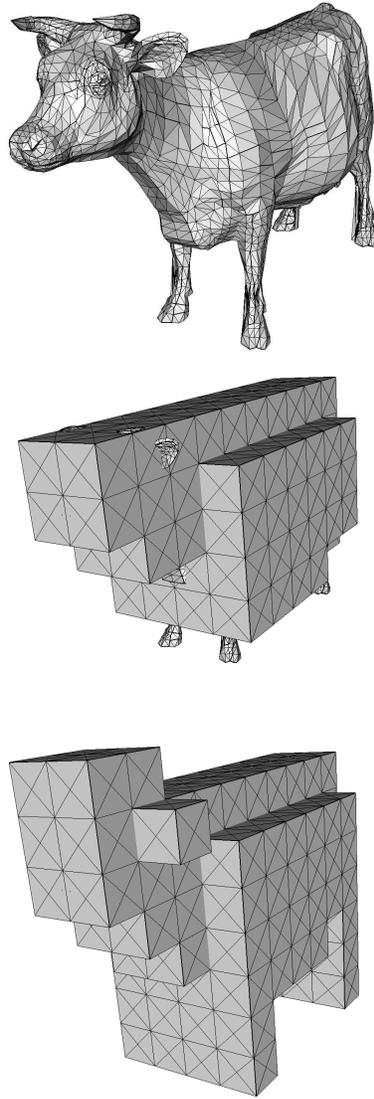


Figure 59: Hidden mixed cell detection for non-smooth geometry. Top figure is original surface mesh, the middle is interface cell detection only by cell vertices, and the bottom shows interface cells after hidden cell check with enriched interior points in the suspicious cells. Hidden mixed cell check resolves accurately the mixed cells for non-smooth shapes even with relatively coarse mesh,  $10^3$  hexahedral mesh.

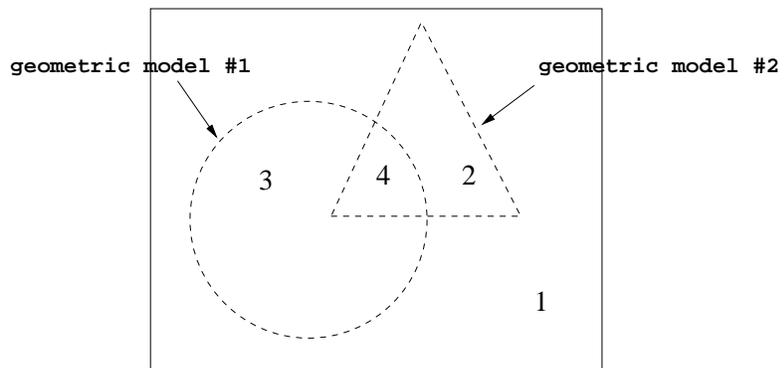


Figure 60: Multi-material configuration for initialization of volume fraction and centroids data. Two different geometric models (circle and triangle) are represented with dashed lines. These two mutually intersecting geometric entities ( $n = 2$ ) can describe up to  $2^n = 4$  unique material regions.

## References

- [1] D. Bailey, S. Brown, and G. Zimmerman. Interface reconstruction and sub-zone physics models. Technical Report UCRL-CONF-214875, Lawrence Livermore National Laboratory, 2005.
- [2] A. J. Barlow. A new lagrangian scheme for multi-material cells. In *Proceeding of European Congress on Computational Methods in Applied Sciences and Engineering - ECCOMAS Computational Fluid Dynamics Conference 2001, Swansea, Wales, UK, 4-7 September 2001.*, 2001.
- [3] D. J. Benson. Computational methods in lagrangian and eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, 99:235–394, 1992.
- [4] D. J. Benson. Eulerian finite element methods for micromechanics of heterogeneous materials: Dynamic prioritization of material interfaces. *Computer Methods in Applied Mechanics and Engineering*, 151:343–360, 1998.
- [5] D. J. Benson. Volume of fluid interface reconstruction methods for multi-material problems. *Appl. Mech. Rev.*, 55:151–165, 2002.
- [6] Yu. Bondarenko and Yu. Yanilkin. Computation of the thermodynamic parameters in the mixed cells in gas dynamics. *Mathematical Modeling*, 14:63–81, 2002.
- [7] K.S. Bonnel, M. A. Ducheneau, D. R. Schikore, B. Hamann, and K. I. Joy. Material interface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 9:500–511, 2003.
- [8] D. E. Burton. Consistent finite-volume discretization of hydrodynamics conservation laws for unstructured grids. Technical Report UCRL-JC-118788, Lawrence Livermore National Laboratory, 1994.
- [9] D. E. Burton. Multidimensional discretization of conservation laws for unstructured polyhedral grids. Technical Report UCRL-JC-118306, Lawrence Livermore National Laboratory, 1994.
- [10] B.Y. Choi and M. Bussmann. A piecewise linear approach to volume tracking a triple point. *International Journal for Numerical Methods in Fluids*, in press:Published online in Wiley Interscience (www.interscience.wiley.com) – DOI:101002/fd.1317, 2006.

- [11] V. I. Delov and V. V. Sadchikov. Comparison of several models for computation of thermodynamical parameters for heterogeneous lagrangian cells. *VANT, Mathematical Modeling of Physical Processes*, 1:57–70, 2005.
- [12] B. Despres and F. Lagoutiere. Numerical resolution of a two-component compressible fluid model with interfaces and mixing zones. Preprint R04008 of the Laboratoire Jacques-Louis Lions.
- [13] J. Donea, S. Giuliani, and J. P. Halleux. An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33:689–723, 1982.
- [14] V. Dyadechko and M. Shashkov. Moment-of-fluid interface reconstruction. Technical Report LA-UR-05-7571, Los Alamos National Laboratory, 2005. <http://cnls.lanl.gov/~shashkov>.
- [15] V. Dyadechko and M. Shashkov. Multi-material interface reconstruction from the moment data. Technical Report LA-UR-06-5846, Los Alamos National Laboratory, 2006. <http://cnls.lanl.gov/~shashkov>.
- [16] D. Fressmann and P. Wriggers. Advection approaches for single- and multi-material arbitrary lagrangian-eulerian finite element procedures. *Comput. Mech.*, 39:153–190, 2007.
- [17] D.M. Gao. A three-dimensional hybrid finite element-volume tracking model for mould filling in casting process. *International Journal for Numerical Methods in Fluids*, 29:877–895, 1999.
- [18] R. Garimella, M. Kucharik, and M. Shashkov. An efficient linearity and bound preserving conservative interpolation (remapping) on polyhedral meshes. *Computers and Fluids*, 36:224–237, 2007.
- [19] Rao Garimella, Milan Kucharik, and Mikhail Shashkov. An efficient linearity and bound preserving conservative interpolation (remapping) on polyhedral meshes. *Computers & Fluids*, 36:224–237, 2007.
- [20] Paul-Louis George. 3D meshes research database. <http://www-c.inria.fr/gamma/gamma.php>.
- [21] Christophe Geuzaine and Jean-Francois Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. <http://www.geuz.org/gmsh/>.

- [22] James Glimm, John W. Grove, Xiao Lin Li, Keh ming Shyue, Yanni Zeng, and Qiang Zhang. Three-dimensional front tracking. *SIAM Journal on Scientific Computing*, 19:703–727, 1998.
- [23] J.A. Harte, W.E. Alley, D.S. Bailey, J.L. Eddleman, and G.B. Zimmerman. Lasnex - a 2d physics code for modeling icf. Technical Report UCRL-LR-105821-96-4, Lawrence Livermore National Laboratory, 1996.
- [24] Michael T. Heath. *Scientific Computing: An Introductory Survey, Second Edition*. McGraw-Hill, New York, 2002.
- [25] C. Hirt, A. Amsden, and J. Cook. An arbitrary lagrangian-eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14:227–253, 1974. reprinted in **135** (1997), pp. 203–216.
- [26] C.W. Hirt and B.D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:201–225, 1981.
- [27] C.T. Kelly. *Iterative methods for optimization*. Society for Industrial and Applied Mathematics, 1999.
- [28] P. Kjellgren and J. Hyvarien. An arbitrary lagrangian-eulerian finite element method. *Computational Mechanics*, 21:81–90, 1998.
- [29] P. Knupp, L. Margolin, and M. Shashkov. Reference jacobian optimization-based rezone strategies for arbitrary lagrangian eulerian method. *Journal of Computational Physics*, 176:93–128, 2002.
- [30] D.B. Kothe, M.W. Williams, K.L. Lam, D.R. Korzekwa, P.K. Tubesing, and E.G. Puckett. A second-order accurate, linearity-preservibg volume tracking algorithm for free surface flows on 3-d unstructured meshes. In *Proceedings of the 3rd ASME/JSME joint fluid engineering conference, San Francisco, Ca. USA, fEDSM99-7109*, 1999.
- [31] P. Liovic, M. Rudman, J.-L. Liow, D.Lakehal, and D. Kothe. A 3d unsplit-advection volume tracking algorithm with planarity-preserving interface reconstruction. *Computers & Fluids*, 35:1011–1032, 2006.
- [32] R. Loubere and M. Shashkov. A subcell remapping method on staggered polygonal grids for arbitrary lagrangian eulerian methods. *Journal of Computational Physics*, 209:105–138, 2005.

- [33] H. U. Mair. Review: Hydrocodes for structural response to underwater explosions. *Shock and Vibration*, 6:81–96, 1999.
- [34] L. Margolin and M. Shashkov. Second-order sign-preserving conservative interpolation (remapping) on general grids. *Journal of Computational Physics*, 184:266–298, 2003.
- [35] L. G. Margolin. Introduction to “an arbitrary lagrangian-eulerian computing method for all flow speeds”. *Journal of Computational Physics*, 135:198–202, 1997.
- [36] D. S. Miller and G. B. Zimmerman. An algorithm for time evolving volume fractions in mixed zones in lagrangian hydrodynamics calculations. Technical Report UCRL-PRES-223908, Lawrence Livermore National Laboratory, 2006.
- [37] G.H. Miller and P.A. Colella. A conservative three-dimensional eulerian method for coupled solid-fluid shock capturing. *Journal of Computational Physics*, 183:26–82, 2002.
- [38] B. Mirtich. Fast and accurate computation of polyhedral mass properties. *Journal of Graphics Tools*, 1:31–50, 1996.
- [39] S. Mosso and S. Clancy. A geometrically derived priority system for youngs’ interface reconstruction. Technical Report LA-CP-95-0081, Los Alamos National Laboratory, 1995.
- [40] S.J. Mosso, B.K. Swartz, D.B. Kothe, and R.C. Ferrel. A parallel, volume-tracking algorithm for unstructured meshes. In P. Schiano, editor, *Parallel Computational Fluid Dynamics ’96, Italy*, 1996.
- [41] A. Murrone and H. Guillard. A five equation reduced model for compressible two phase flow problems. *Journal of Computational Physics*, 202:664–698, 2005.
- [42] Joseph O’Rourke. *Computational Geometry in C*. Cambridge University Press, 2001. Second edition.
- [43] S. Osher and R.P. Fedkiw. Level set methods: An overview and some recent results. *Journal of Computational Physics*, 169:463–502, 2001.
- [44] J. S. Peery and D. E. Carroll. Multi-material ale methods in unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 187:591–619, 2000.

- [45] J.E. Pilliod and E.G. Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics*, 199:465–502, 2004.
- [46] E.G. Puckett. A volume-of-fluid interface tracking algorithm with applications to computing shock wave refraction. In H. Dwyer, editor, *Proceedings of the Fourth International Symposium on Computational Fluid Dynamics*, pages 933–938, 1991.
- [47] E.G. Puckett and J.S Saltzman. A 3d adaptive mesh refinement algorithm for multimaterial gas dynamics. *Physica D*, 60:84–93, 1992.
- [48] W.J. Rider and D.B. Kothe. Reconstructing volume tracking. *Journal of Computational Physics*, 121:112–152, 1998.
- [49] M. Rudman. Volume tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids*, 24:671–691, 1997.
- [50] R. Scardovelli and S Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics*, 31:567–603, 1999.
- [51] S.P. Schofield, R.V. Garimella, M.M. Francois, and R. Loubere. Mutli-material interface reconstruction using particles and power diagrams. Technical Report LA-UR-06-8740, Los Alamos National Laboratory, 2006.
- [52] A. Stagg, R. Boss, J. Grove, and N. Morgan. Interface modeling: A survey of methods with recommendations. Technical Report LA-UR-05-8157, Los Alamos National Laboratory, 2005.
- [53] Michael B. Stephenson and Henry N. Christiansen. A polyhedron clipping and capping algorithm and a display system for three dimensional finite element models. *ACM SIGGRAPH Computer Graphics*, 9:1–16, 1975.
- [54] H. Stewart and B. Wendroff. Two-phase flow: Models and methods. *Journal of Computational Physics*, 56:363–409, 1984.
- [55] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome. An adaptive level set approach for incompressible two-phase flows. *Journal of Computational Physics*, 148:81–124, 1999.

- [56] M. Sussman, E. Fatemi, P. Smereka, and S. Osher. Improved level set method for incompressible two-phase flows. *Computers & Fluids*, 27:663–680, 1998.
- [57] B. Swartz. The second-order sharpening of blurred smooth borders. *Math. Comput.*, 52:675–714, 1989.
- [58] The Telluride Team. TRUCHAS physics and algorithms. Technical Report LA-UR-06-1379, Los Alamos National Laboratory, 2006.
- [59] R. E. Tipton. Cale mixed zone pressure relaxation model, 1989. Private Communication.
- [60] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahic, W. Tauber, J. Han, S. Nasr, and Y. J. Jan. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics*, 169:708–759, 2001.
- [61] S.O Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100:25–37, 1992.
- [62] Efrem Vitali and David J. Benso. An extended finite element formulation for contact in multi-material arbitrary lagrangian-eulerian calculations. *International Journal for Numerical Methods in Engineering*, 67:1420–1444, 2006.
- [63] Xiaofeng Yang and Ashley J. James. Analytic relations for reconstructing piecewise linear interfaces in triangular and tetrahedral grids. *Journal of Computational Physics*, 214:41–54, 2006.
- [64] D.L. Youngs. *Numerical Methods for Fluid Dynamics*, chapter Time-dependent Multi-material Flow with Large Fluid Distortion, pages 273–285. Academic Press:New York, 1982.