Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 226 (2007) 2096–2132
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Multi-material interface reconstruction on generalized polyhedral meshes

Hyung Taek Ahn *, Mikhail Shashkov

Theoretical Division, T-7, Los Alamos National Laboratory, MS-B284, Los Alamos, NM 87545, USA

Received 11 April 2007; received in revised form 26 June 2007; accepted 28 June 2007
Available online 26 July 2007

Abstract

We describe multi-material (more than two materials) interface reconstruction methods for 3D meshes of generalized polyhedra. The basic information used in interface reconstruction is the volume fraction of each material in mixed cells, that is, those containing multiple materials. All methods subdivide a mixed cell into a set of pure non-overlapping sub-cells, each containing just one material that have the reference volume fraction. We describe three methods. The first two methods represent an extension of standard piece-wise linear interface construction (PLIC) methods to 3D and use information only about volume fractions. The first method is first-order accurate and is based on the discrete gradient of the volume fraction as an estimate of the normal to the interface. The second method is planarity-preserving (second-order accurate) and is an extension to 3D of the least squares volume-of-fluid interface reconstruction algorithm (LVIRA, see [E. Puckett, A volume-of-fluid interface tracking algorithm with applications to computing shock wave refraction, in: H. Dwyer (Ed.), Proceedings of the Fourth International Symposium on Computational Fluid Dynamics, 1991, pp. 933–938; J.E. Pilliod, E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, Journal of Computational Physics 199 (2004) 465–502] for the 2D case). The third method is an extension to 3D of the so-called moment-of-fluid (MoF) method [V. Dyadechko, M. Shashkov, Moment-of-fluid interface reconstruction, Tech. Rep. LA-UR-05-7571, Los Alamos National Laboratory, 2005. Also available as http://cnls.lanl.gov/~shashkov/; V. Dyadechko, M. Shashkov, Multi-material interface reconstruction from the moment data, Tech. Rep. LA-UR-06-5846, Los Alamos National Laboratory, 2006. Also available as http://cnls.lanl.gov/~shashkov/]. The MoF method is also second-order accurate. This method uses information not only about volume fractions but also about the position of the centroids of each material. In contrast to standard PLIC methods, the MoF method uses only information from the cell where reconstruction is performed; no information from neighboring cells is needed. Also, the MoF method provides automatic ordering of the materials during interface reconstruction. Optimal ordering is based on comparing the positions of the reference centroids and actual centroids of the reconstructed pure sub-cells. The performance of the methods is demonstrated with numerical examples.
Published by Elsevier Inc.

Keywords: Interface reconstruction; Multi-material flow; Multi-phase flow; Polyhedral mesh

* Corresponding author. Tel.: +1 505 665 9188; fax: +1 505 665 5757.
E-mail addresses: htahn@lanl.gov (H.T. Ahn), shashkov@lanl.gov (M. Shashkov).

0021-9991/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jcp.2007.06.033

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 [5], Hirt et al. developed the formalism for a grid whose motion could be determined as an independent degree of freedom, and they 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 [6–11].

It is possible to formulate the ALE scheme as a single algorithm [12,11] based on solving the equations in a moving coordinate frame. For multi-material flows it is usual to separate the 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 meshes 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 [13–16].

For multi-material flows, the initial mesh is usually aligned with material interfaces; 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 interfaces aligned with the mesh, that is, the rezoning does not move nodes on the interface at all, or moves them 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, so it is unavoidable that mixed cells containing two or more materials will appear.

Multi-material cells in ALE methods include 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 how to accurately determine 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 required to define how the volume fractions and states of the individual materials evolve during the Lagrangian step [17–24]. 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 recent methods like those presented in [17,20] do require interface locations and orientations and therefore require some representation of the interface.

In the rezoning stage, the nodes of the computational grid are moved to a more optimal position. For the purpose of this paper it is not important what algorithm is used. Interested readers can find a review of rezoning algorithms in [25]. The rezoning stage results in a new mesh. To start the next Lagrangian step, we need to conservatively interpolate all flow parameters from the Lagrangian mesh at the completed time step to the new rezoned mesh. This process is called remapping.

Remapping for the case when there is only one material (and therefore no mixed cells) is described, for example, in [26–28]. In the situation when several materials are present, even if after the Lagrangian step all cells were pure (containing only one material), the rezoned mesh will have mixed cells. Therefore, in the remap stage one needs to determine which cells of the rezoned mesh are pure and which are mixed, and in particular, to find parameters for each material in mixed cells. There are several approaches to perform

multi-material remapping, but all of them require knowledge of the interface in the Lagrangian mesh after each completed time step. Therefore, the interface has to be represented in some way on the mesh after a Lagrangian step.

There are several well established methods for dealing with interfaces: the volume-of-fluid (VoF) method (which uses interface reconstruction) [29–31]; the front tracking [32–34]; and level set method [35–37].

For modeling three-dimensional, high-speed compressible, multi-material flows, on general meshes with the interface topology changing in time, and when exact conservation is critical, VoF seems to be method of choice [31,38,6,39].

Originally, VoF was developed for modeling the dynamics of incompressible flows with free boundaries using an Eulerian approach, [29]. The basic object of VoF methods is a *two material medium*. The typical VoF method consists of two steps: interface reconstruction (using volume fractions) and updating the volume fractions in time. Excellent reviews of VoF methods and, in particular, general interface reconstruction methods can be found in the following papers [30,31,2,40,41]. 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 (a line in 2D and a plane in 3D) per mixed cell containing two materials. This class of interface representation is commonly called Piecewise-Linear Interface Calculation (PLIC). The location of the linear interface, for a given volume fraction, is uniquely defined by the direction of the interface outward normal. There are a number of ways to define the direction of the normal [3]. We will describe some of them in Section 3.1.

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

The standard VoF interface reconstruction methods for two materials use only information about volume fractions. Recently, we have developed a new method, the *moment-of-fluid* or MoF method [3,4], which also uses information about the position of the centroid of each material in the mixed cell. In the MoF method, the objective is to minimize the distance between specified (reference) and actual centroids obtained as a result of reconstruction, while exactly matching the specified volume fraction. In 2D it leads to minimization of a one-dimensional function [3].

Here we want to mention that there are hydrocodes where centroids of the materials are tracked, see for example [42,43] for the LASNEX code description, and there are algorithms for transporting centroids in finite element hydrocodes in [31,38].

For mixed cells containing more than two materials, we are only interested in algorithms where each material will be represented by a collection of pure sub-cells (possibly disjoint). These sub-cells do not intersect with each other, and the union of all sub-cells representing all materials is equal to the entire mixed cell. The total volume of all sub-cells representing material of one kind has to match the corresponding volume fraction. It is interesting to know that in most VoF algorithms the interface reconstruction procedure does not satisfy this

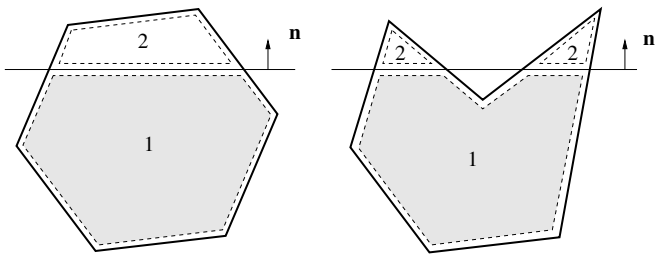

Fig. 1. Two material interface reconstruction in 2D. Left shows a convex polygonal cell, and right corresponds to the non-convex case. If the original cell is non-convex, the sub-cells after intersection remain non-convex.

property [31,44], because they use the *onion skin* model [45,31], which may lead to intersection of sub-cells representing different materials. The problems related to this phenomenon are usually resolved at the advection stage of VoF methods by limiting the fluxes [44,46]. More recent methods produce sub-cells representing different materials which do not intersect each other [47,48,4].

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

To the best of our knowledge there are only two approaches where multi-material interface reconstruction does not require ordering. The first method introduced in [49] uses an innovative extension of a contouring algorithm. This method does not reproduce specified volume fractions exactly. The second method, based on the use of particles and power diagrams [50], is very promising but requires more work to be considered for practical applications. We do not consider these methods in our paper.

Following [4], we will refer to an algorithm that is based on sequential application of an algorithm for two materials, and which satisfies our requirements, as *nested dissection*.

The general idea that materials are cut from a mixed cell sequentially in a specified order is illustrated in Fig. 2. Let us denote the mixed cell by Ω. The first material #1 is separated from the rest of the materials in the mixed cell. It can be done by any two material interface reconstruction method, where the first material is material #1 and the second material is the union of all other materials. This first step creates a pure sub-cell (or several pure sub-cells in case of a non-convex mixed cell) which represent the first material. We will denote this set by Ω_1. After this step material #1 is removed from the mixed cell, and we now consider the set $\Omega\backslash\Omega_1$. This set contains materials $\#2, 3, \ldots$ The next step is to remove material #2 from set $\Omega\backslash\Omega_1$. This again can be done by applying the same two material algorithm to $\Omega\backslash\Omega_1$, where the first material will be material #2 and the second material will be the union of materials $\#2, 3, \ldots$ The volume fraction of material #2 has to be recomputed with respect to the set $\Omega\backslash\Omega_1$. This second step creates pure sub-cells representing material #2, denoted by Ω_2. After this step, material #2 is removed from the mixed cell, and we consider the set $\Omega\backslash\Omega_1\backslash\Omega_2$. This process is repeated until we create pure sub-cells for all materials.

There have been several attempts to develop automatic-local schemes for material ordering [51,47,38,43] for standard PLIC methods. All of them are based on using information about the exact or approximate centroid positions of the materials in the mixed cell itself and its neighbors. To the best of our knowledge, e.g examples in [4], none of these methods treat even all basic cases correctly.

In the MoF method, the positions of the material centroids inside the mixed cell are used not only for each two material interface reconstruction step, but also for choosing the best material ordering that minimizes the discrepancy between the given (reference) and actual centroids computed from the reconstructed pure sub-cells for all materials [4].

A recent paper [52] deals with a special case of a triple point configuration in 2D on a square mesh. The algorithm presented in [52] is very specialized and assumes that the first material in the reconstruction process is externally specified; the position of the second interface is obtained from an optimization procedure.

Most of the papers related to interface reconstruction deal with the two-dimensional case. There are only several papers which describe interface reconstruction in 3D; almost all of them deal with a brick mesh and the case of two materials, e.g. [53], and references therein. In [53] a 3D extension of the *Swartz–Mosso*, algorithm

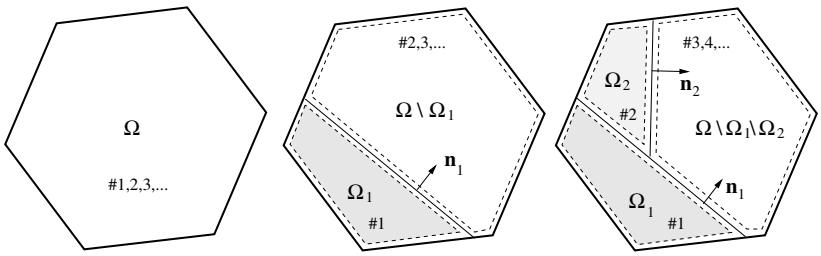

Fig. 2. Nested dissection strategy on a 2D polygonal cell. Three consecutive stages of nested dissection: left – three material mixed cell; middle – after first dissection; right – after second dissection.

[54,55,3] is developed, but it is described only for an orthogonal mesh. However, the authors of [53] claim that it can be easily extended to unstructured and distorted meshes. Descriptions of 3D interface reconstruction methods on distorted and unstructured meshes are very rare and are usually in unpublished reports related to ALE methods, papers published in proceedings of conferences, e.g. [56], or in some papers related to specific applications, e.g. [57]. We refer an interested reader to [46] 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 the MoF method for multi-material interface reconstruction to the case of 3D meshes consisting of generalized polyhedra. We also present possible extensions of two standard methods for interface reconstruction for two materials to the case of multi-material interface reconstruction on a 3D mesh consisting of generalized polyhedra. One method is based on estimation of the gradient of the volume fraction, and the other is the least squares volume-of-fluid interface reconstruction algorithm (LVIRA) [1,2].

The outline of the rest of this paper is as follows. In Section 2, we describe a class of 3D meshes consisting of generalized polyhedra and corresponding data structures. In next Section 2.2, we describe how materials are represented in a multi-material mixed cell. In Section 3, we give a general description of three interface reconstruction methods for the case of two materials: the volume fraction gradient based (GRAD) method, LVIRA and MoF, and we present numerical examples for two material interface reconstruction. Geometric algorithms needed for interface reconstruction, optimization algorithms for LVIRA and MoF, and initialization of volume fraction and centroid data are described in the Appendices of [58]. In Section 4, we describe extensions of GRAD, LVIRA and MoF methods to the 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 meshes

2.1. Definition of the generalized polyhedral mesh (GPM)

In this paper, we are interested in a mesh that consists of *generalized polyhedra – generalized polyhedral mesh* (GPM). A generalized polyhedron can be thought as a 3D solid obtained from a polyhedron by perturbing positions of its vertices, which makes its faces non-planar. As we have mentioned in Section 1, generalized polyhedra naturally appear in ALE calculations, because even if the initial mesh consists of classical polyhedra such as hexahedra with planar faces at the start of the simulation, after a 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 (see [28] for more detail). First, we define the "face center"

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

where $\mathcal{P}(f)$ is the set of all face vertices, $|\mathcal{P}(f)|$ is the number of vertices for the face, and $\mathbf{x}_c^{\text{face}}$ and $\mathbf{x}_i$ are the coordinates of the face center and the ith vertex, respectively. Next, the faces of the generalized polyhedral cell are 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 surfaces and one additional vertex inside the cell, called the "cell center", defined as follows:

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

where $\mathcal{P}(c)$ is the set of all cell vertices, $|\mathcal{P}(c)|$ is the total number of vertices for the cell, and $\mathbf{x}_c^{\text{cell}}$ and $\mathbf{x}_i$ are the coordinates of the cell center and the ith vertex, respectively. Hence, an m-vertexed polygonal face is divided into m-triangles, and an n-faced polyhedron is further decomposed into $n \times m$ sub-tetrahedra provided that each face is composed of m vertices. For example, the generalized polyhedral representation of a hexahedral cell results in 6×4 sub-tetrahedra as displayed in Fig. 3.

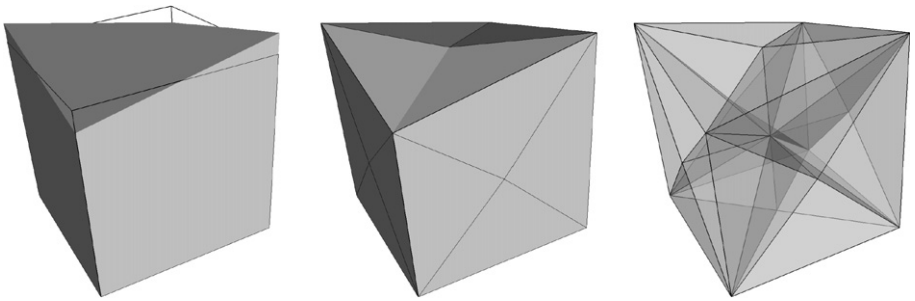

Fig. 3. Generalized polyhedral representation of a hexahedron with non-planar faces. The left figure shows the initial hexahedral cell with a non-planar top face (the wire frame of a cube is overlapped to emphasize the non-planar top face), the middle figure shows the surface triangulation of the hexahedron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

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

This generalization of polyhedral cells has two advantages. First, the planar face restriction of a polyhedral cell is relaxed so that it can have vertices not always in 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 (that is, the cell center $\mathbf{x}_c^{\text{cell}}$ 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 the previous section, each cell of the mesh in 2D is decomposed into convex sub-cells (triangles) as shown in Fig. 4. A general scheme for multi-material interface reconstruction is described in the Introduction and consists in nested dissection of the original cell. The first step of this process is to define the part of the cell which is occupied by material #1. The cell is intersected with a half-plane (specified by normal $\mathbf{n}$ and signed distance d), such that this intersection has the prescribed volume (volume fraction) (Fig. 5). In Fig. 5, the part of the cell occupied by material #1 is shaded. Technically, material #1 is represented as a collection of shaded polygons, each of which is part of one of the triangles of the original triangulation of the cell. These polygons are intersections of a half-plane with the corresponding triangles. The sum of the volumes of these polygons matches the 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 specified for each triangle.

Materials #2, 3, ... are contained in non shaded polygons which are also parts of triangles of the original triangulation. The nested dissection scheme continues to dissect non shaded polygons to find the part of the

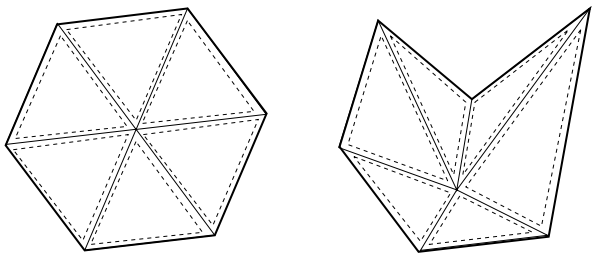

Fig. 4. Representation of the polygonal cell in 2D. The sub-cell (indicated by dashed lines) structure of *pure* generalized polygonal cell is illustrated for the convex (left) and non-convex (right) case. Notice that sub-cells are always convex regardless of convexity of the original cell.

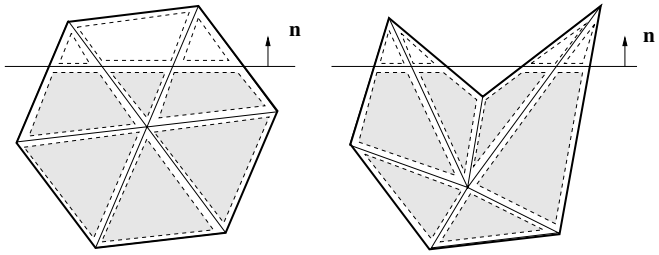

Fig. 5. Two material mixed cell in 2D. The left represents a convex cell and the right figure represents the non-convex case. Notice that all sub-cells are always convex regardless of convexity of the original cell or the presence of material interface.

cell which represents material #2. It easy to see that material #2 will be represented by a collection of convex polygons which are intersections of a corresponding half-plane with the non shaded polygons obtained in the first step. To find convex polygons representing material #3, one needs to repeat the process.

In 3D, the process is the same: one just needs to substitute triangles with tetrahedra, and half-planes by half-spaces, as illustrated in Fig. 6. Three different types of generalized polyhedral mixed cells are represented in Fig. 6. Each cell includes three materials (colored red, green and blue). An example of a non-convex enneahedron, displayed in the middle row of Fig. 6, demonstrates the case where one material (blue) is represented by disjoint pieces. Similarly to the 2D case, each interface is reconstructed by the intersection of the polyhedral cell with half-spaces. The sub-cells, initially tetrahedra, evolve to convex polyhedra as they intersect with their corresponding half-spaces. A wire frame view of these sub-cells, shown in the left column of Fig. 6, reveals these sub-cell structures.

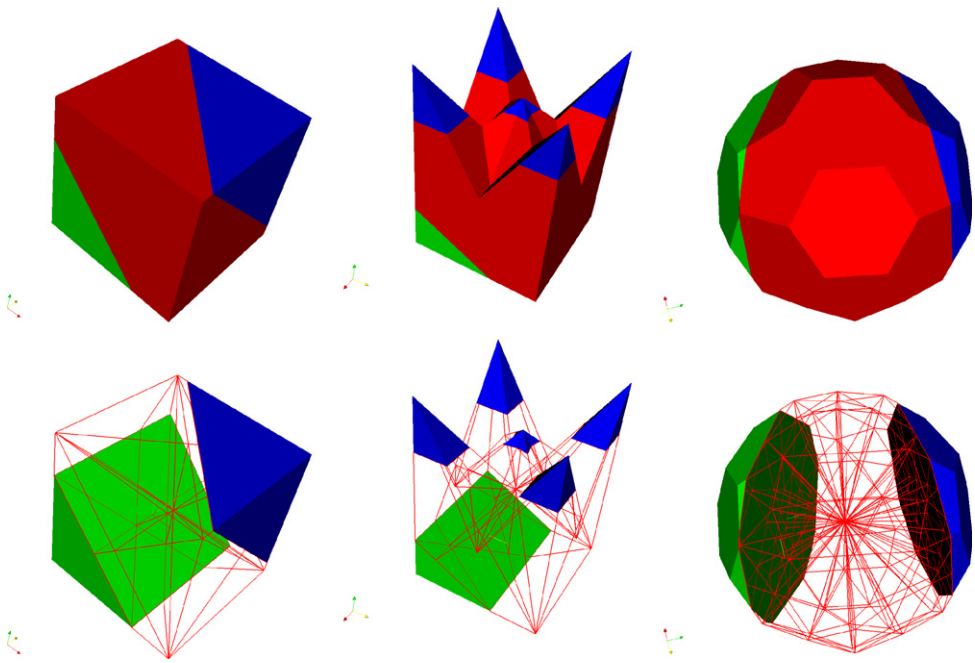

Fig. 6. Generalized polyhedral cells with multi-materials (red, green and blue). The left column shows a hexahedral cell, the middle column shows a *non-convex* enneahedron (obtained by subdividing the top face of a hexahedron and disturbing the vertices on the faces) and the right column represents a truncated icosahedron. The top row shows the solid view and the bottom row shows the wire-frame view of the solid which reveals the sub-cell structure. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

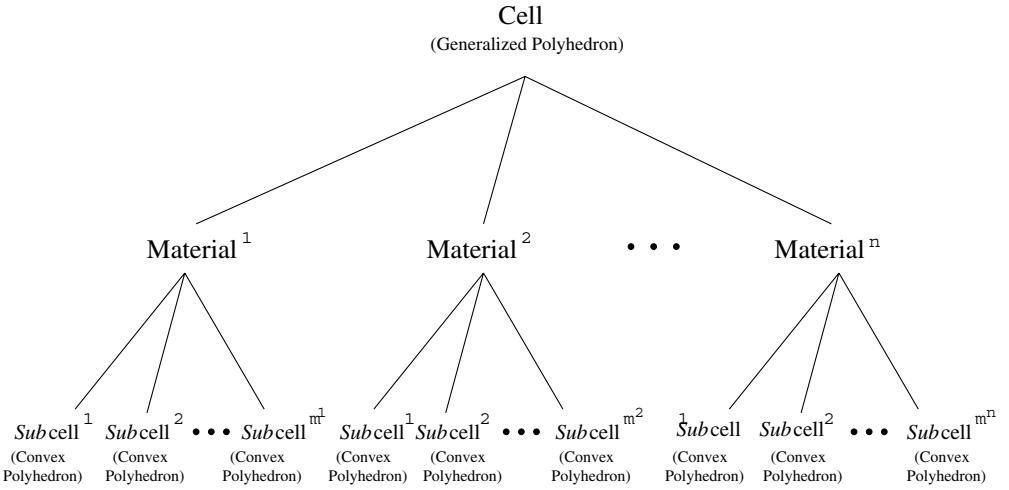

Fig. 7. Hierarchical structure of a generalized polyhedral cell (which may be non-convex) containing multiple materials. Each material is represented by a set of non-overlapping *pure* sub-cells which are always convex.

In summary, each material in the mixed cell is represented as a collection of convex polyhedra. The hierarchical structure of a mixed cell is delineated in Fig. 7. Each of these polyhedra is obtained as a result of the intersection of one of the original tetrahedra with half-spaces.

3. Interface reconstruction for two materials

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

3.1. Review of representative PLIC methods for the case of two materials

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

$$\mathbf{n} \cdot \mathbf{r} + d = 0 \tag{3}$$

where $\mathbf{r} = (x, y, z)$ is a 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 the cell, c, with a volume equal to the reference volume V_c^{ref} of the material (or equivalently, the volume fraction $f_c^{\text{ref}} = V_c^{\text{ref}}/V_c$, where V_c is the volume of the entire cell c).

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

Most of the standard PLIC methods [2], use information only about the volume fraction, f_c^{ref}, in the cell c under consideration and volume fractions, $f_{c\prime}^{\text{ref}}$, in the neighboring cells $c\prime \in \mathcal{C}(c)$ (here $\mathcal{C}(c)$ is the set of cells which share a face, edge, or vertex with cell c). In our study, standard PLIC methods are represented with two methods, which can be naturally extended to 3D meshes of generalized polyhedra. The first method, GRAD (see Section 3.1.1 for more detail), is an extension of *the central difference algorithm* and Parker and Youngs' method as described in [2]. This method uses an estimate of the gradient of the volume fraction field to

compute a normal to interface. The second method is an extension of the least squares volume-of-fluid interface reconstruction algorithm (LVIRA), [1,2]. Here, one attempts to find a planar interface in a mixed cell, which, being extended into neighboring cells, gives the best (in a least squares sense) match to the reference volume fractions in these cells with the constraint that this plane exactly reproduces the volume fraction in the cell under consideration (see Section 3.1.2 for detail). Both the GRAD and LVIRA methods use only information about volume fractions.

The third method is the *moment-of-fluid* (MoF) method which was introduced in [3]. The MoF method uses information about volume fractions and centroids, but *only* the cell in consideration. Information from neighboring cells is not used. In the MoF method, the plane representing the interface is chosen to match the reference volume exactly and to provide the best approximation (in a least squares sense) to the reference centroid of the material (see Section [3] for detail).

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

3.1.1. Gradient based interface reconstruction (GRAD)

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

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

In the case of a 3D unstructured mesh consisting of generalized polyhedra, it is convenient to use a least squares procedure (see, for example [28]) to estimate the gradient of the volume fraction. For each cell c, a linear function $f_c(\mathbf{r})$ can be reconstructed in the following form:

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

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 the components of the volume fraction gradient, $\alpha_c = \int_c \alpha \,\mathrm{d}V / V_c$ are the coordinates of the cell centroids and $V_c = \int_c 1 \,\mathrm{d}V$ is the cell volume. The actual computation of the centroids and the volume of the polyhedral cell is based on the algorithm introduced in [59]. Estimates for the components of the gradient can be found from minimization of the following functional:

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

Solution of the least squares minimization problem reduces to solving a 3×3 system of linear equations.

The resulting normal $\mathbf{n}$,

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

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

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

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

$$\mathcal{V}(d) = V^{\mathrm{ref}} \tag{8}$$

The volume $\mathcal{V}(d)$ is a continuous and monotone function of d, which guarantees that Eq. (8) always has a unique solution. The procedure which we use in this paper to solve Eq. (8) is described in the Appendix B

of [58], and it is based on secant and bisection methods. This procedure requires the computation of the volume of the intersection of a half-space and the true polyhedron, which is also described in [58]. Let us note that all PLIC methods require solving Eq. (8) (maybe several times), geometrical algorithms for the intersection of a half-space and a convex polyhedron, and computation of the volume of a polyhedron.

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

Therefore, it is not surprising that the GRAD method in 3D on GPM is incapable of recovering planar interfaces as shown in Fig. 8. According to [2], it indicates the first-order accuracy of the GRAD method.

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

In the LVIRA interface reconstruction method introduced by Puckett and co-workers [1,2], the interface normal is computed by minimizing the following error functional:

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

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

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

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

In contrast with the GRAD method, LVIRA requires minimization of the non-linear objective function, as shown in Eq. (9). In 3D, the normal can be described by polar angles, and therefore implementation of LVIRA requires an algorithm for the minimization of a non-linear function of two variables. In this paper we use the iterative Broyden–Fletcher–Goldfarb–Shanno (BFGS) method (see Appendix C of [58] for details).

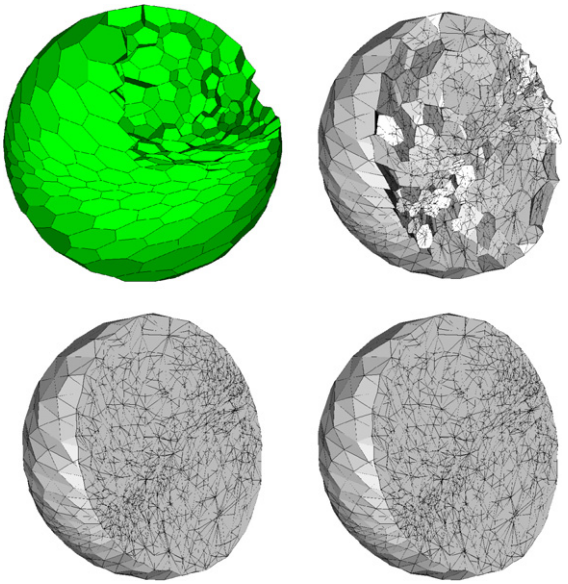

Fig. 8. Interface reconstruction of a planar interface on a generalized polyhedral mesh. The top-left figure shows the original polyhedral mesh (the subdivision of the cells into tetrahedra is not shown) obtained from [28]. The top-right figure shows a GRAD reconstruction, which is a first-order accurate method. The bottom-left figure shows an LVIRA reconstruction and the bottom-right figure shows a MoF reconstruction, which are both planarity-preserving (second-order) accurate methods.

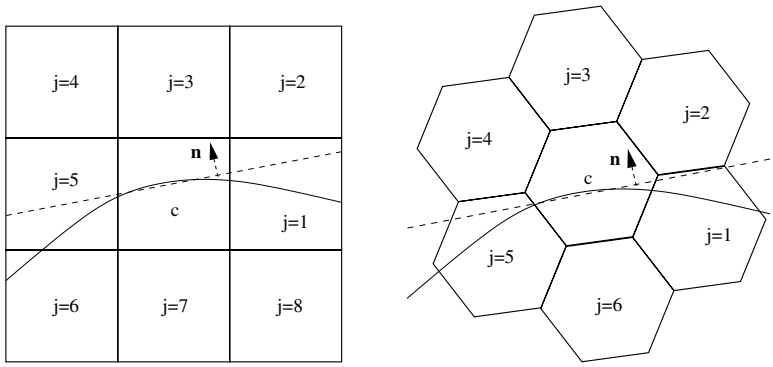

Fig. 9. Stencil for LVIRA error computation in two dimensions. The stencil is composed of immediate vertex neighbors of the central cell-c. The solid curved line represents the true interface and the dashed straight line represents the extension of a piecewise linear volume fraction matching interface at the central cell-c. The left picture represents a structured quadrilateral mesh and the right picture represents an unstructured polygonal mesh.

The initial guess for the normal is given by the GRAD method. Implementation of this algorithm requires computation of $E_c^{\mathrm{LVIRA}}(\mathbf{n})$. To do this for a given normal $\mathbf{n}$, we first need to find the parameter d of the plane such that the volume fraction in cell c exactly matches f_c^{ref}, which is the step that is performed in the GRAD method. Then the resulting plane is extended into neighboring cells and the intersection of the corresponding half-space with each neighboring cell c' is used to compute actual volume fractions $f_{c'}(n)$. Next, these volume fractions are used to compute the error functional $E_c^{\mathrm{LVIRA}}(\mathbf{n})$, expressed in Eq. (9). We will note that all of these operations have to be performed on each iteration of the minimization procedure.

With respect to the GRAD method, one new algorithm is required – an algorithm for the non-linear optimization of functions of two variables.

Clearly, LVIRA 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 26 polyhedral cell intersections for 3D hexahedral meshes.

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

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

The *moment-of-fluid* method was introduced by Dyadechko and Shashkov [3] for interface reconstruction in 2D. The MoF method uses information about the volume fraction, f_c^{ref} and centroid, $\mathbf{x}_c^{\mathrm{ref}}$ of the material, but *only* from the cell c under consideration. No information from neighboring cells is used, as illustrated in Fig. 10.

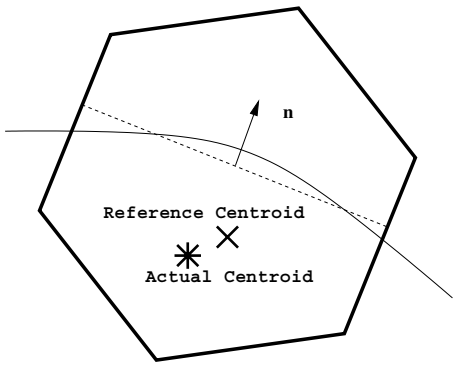

Fig. 10. Stencil for MoF error computation in two dimensions. The stencil is composed of only the cell under consideration. The solid curved line represents the true interface and the dashed straight line represents the piecewise linear, volume fraction matching interface at the cell.

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

$$E_c^{\mathrm{MoF}}(\mathbf{n}) = \left\|\mathbf{x}_c^{\mathrm{ref}} - \mathbf{x}_c(\mathbf{n})\right\|^2 \tag{10}$$

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

Similar to the LVIRA, the implementation of MoF method requires the minimization of the non-linear function of two variables; thus we use the same BFGS algorithm as that for LVIRA optimization. It is shown in [3] that the vector from the reference centroid to the centroid of the entire cell is a good initial guess for the interface normal, and we use that guess in this paper.

The computation of $E_c^{\mathrm{MoF}}(\mathbf{n})$ requires the following steps. The first step is to find the parameter d of the plane such that the volume fraction in cell c exactly matches f_c^{ref}; this is also performed in the GRAD and LVIRA algorithms. Secondly, we compute the centroid of the resulting polyhedron. This is a simple calculation, described in [59]. Finally, one computes the distance between actual and reference centroids.

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

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

3.2. Numerical examples

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

The initialization method for volume fraction and moment data is described in Appendix D of [58].

3.2.1. Smooth interface

In this section, we present a mesh convergence study for the GRAD, LVIRA and MoF interface reconstruction methods based on an example of reconstructing a spherical shape. The sphere is centered at $\left(0.5+\frac{1}{29}, 0.5+\frac{1}{31}, 0.5+\frac{1}{39}\right)$ 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. 11, we present results of interface reconstruction using four levels of successively refined hexahedral meshes.

The first column of Fig. 11 shows the behavior of the GRAD method. The gradient based scheme shows convergence to the true spherical interface as mesh is refined, however, the discontinuity of surfaces between cells persists even for the highest mesh resolution. This confirms that the gradient based scheme exhibits an interface normal error which is independent of local mesh size, h. Hence, the scheme is first-order accurate.

The second column of Fig. 11 displays LVIRA interface reconstruction. This method is formally second-order accurate but demonstrates the worst performance on mesh of lowest 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 the reconstructed interface appear more and more continuous. Results obtained by LVIRA appear better than those obtained by the GRAD method.

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

A quantitative measure of the errors in interface reconstruction is presented in Fig. 12. The error is measured as the volume of the symmetric difference, defined as follows:

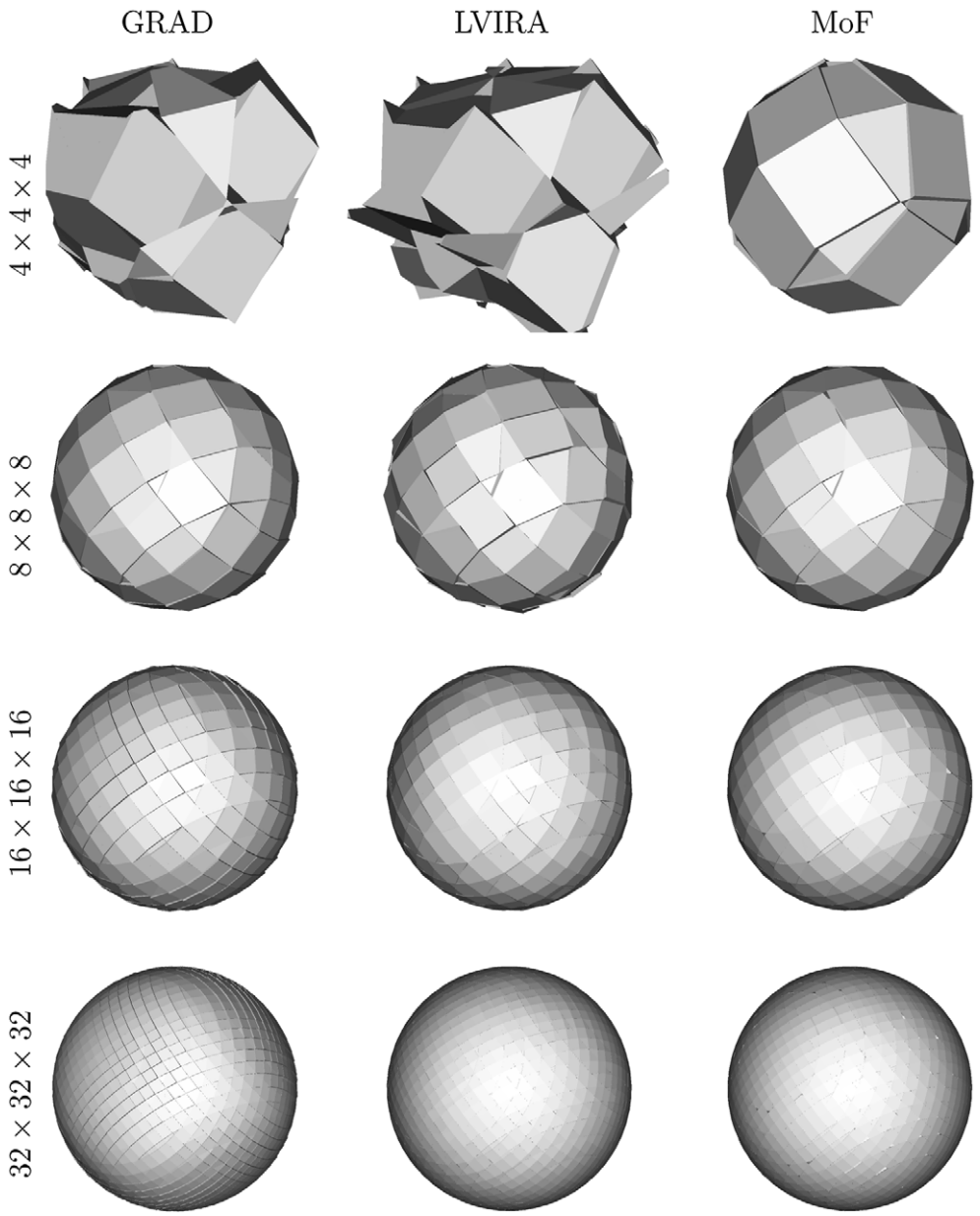

Fig. 11. Mesh convergence study with a spherical material region. Equispaced hexahedral meshes covering the domain of $[0,1]^3$ are employed.

$$T \triangle R = (T \cup R) - (T \cap R) \tag{11}$$

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

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

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

The convergence rate of the reconstruction is displayed in Fig. 12. The results demonstrate that: (i) the gradient based scheme is only first-order accurate, (ii) LVIRA can be less accurate than the GRAD method on coarse meshes but shows second-order accuracy on fine meshes and (iii) MoF is second-order accurate and is the most accurate for all levels of mesh resolution.

3.2.2. Interfaces with sharp corners

There are realistic situations where sharp interfaces are present and accurate reconstruction of such interfaces is important. For example, consider sharp corners in a contact problem or triple points in multi-material flow simulations.

Here, we present a mesh convergence study using 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 an angle of $\alpha = \pi/6$. The reconstruction of the material region using equispaced hexahedral meshes is displayed in Fig. 13. As also noted in Fig. 8, the first-order accurate GRAD method is not capable of recovering flat interface exactly, but the second-order accurate methods, LVIRA and MoF, preserve planarity. Due to its local nature, the MoF method best resolves the sharp edges of the cube.

Errors from interface reconstruction are measured by the volume of the symmetric difference between the true and reconstructed material regions. The rate of convergence is displayed in Fig. 14. Both LVIRA and MoF show second-order accuracy, but GRAD deviates towards first-order accuracy as mesh refines. For all mesh resolutions, MoF shows the most accurate results.

For another test case of two material interface reconstruction, we consider interfaces with sharp corners including a not simply connected material region, shown in Fig. 15.

Fig. 15 shows the results of the three interface reconstruction methods for this test. The original surface mesh is obtained from [60]. First, the GRAD method and LVIRA results show significant smoothing along the sharp corners. This is due to their dependency on neighboring cells, i.e. non-locality. Both methods need information from neighbors either because of a gradient computation or due to volume fraction error contributions. On the contrary, the MoF interface reconstruction results shows well resolved corners within the tolerance of a single cell due to the independence of the method from calculations using the neighbor cells, i.e. locality.

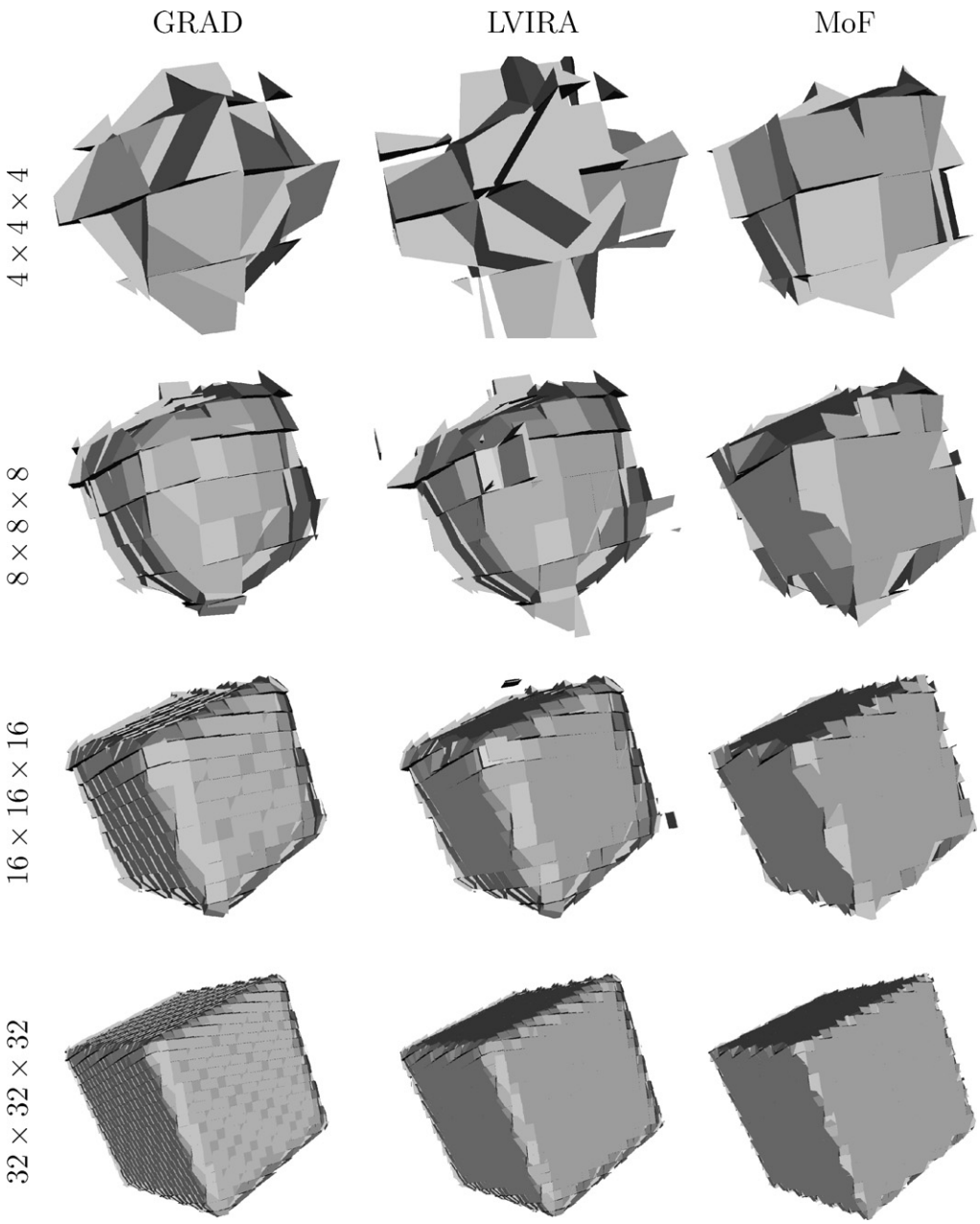

Fig. 13. Mesh convergence study with a cubical material region. Equispaced hexahedral meshes covering the domain of $[0,1]^3$ are employed.

3.2.3. Reconstruction on an unstructured mesh

We now consider an unstructured tetrahedral mesh. A spherical material region centered at $(0.5, 0.5, 0.5)$ with radius $r = 0.4$ is considered. The unstructured tetrahedral mesh is generated with Gmsh [61]. The reconstruction of the spherical material region on the unstructured mesh is displayed in Fig. 16. The left-top figure shows the subcell decomposed tetrahedral mesh, and top-right figure shows GRAD, the bottom-left figure for LVIRA result, and the bottom-right shows MoF results. The inter-cell interface discontinuity is minimal in MoF reconstruction, which highlights the superiority of the MoF method.

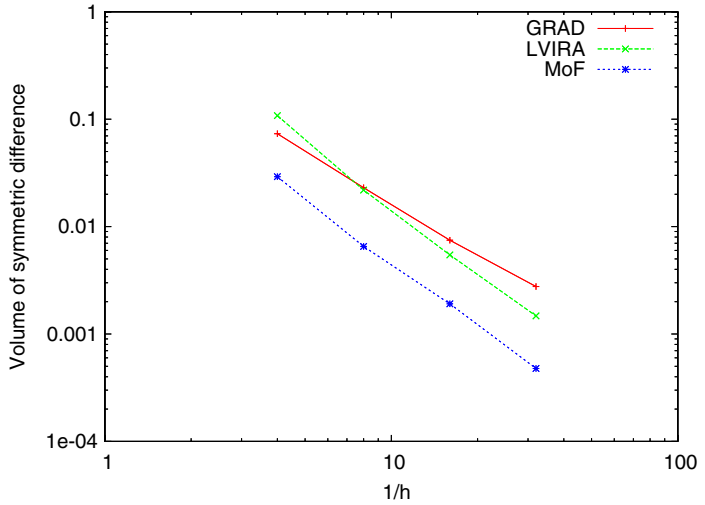

Fig. 14. Convergence of reconstruction error measured as the volume of the symmetric difference between true and reconstructed material regions for the cubical material test case. The convergence rates at the last refinement step are: $p^{\mathrm{GRAD}} = 1.433$, $p^{\mathrm{LVIRA}} = 1.885$, $p^{\mathrm{MoF}} = 2.022$.

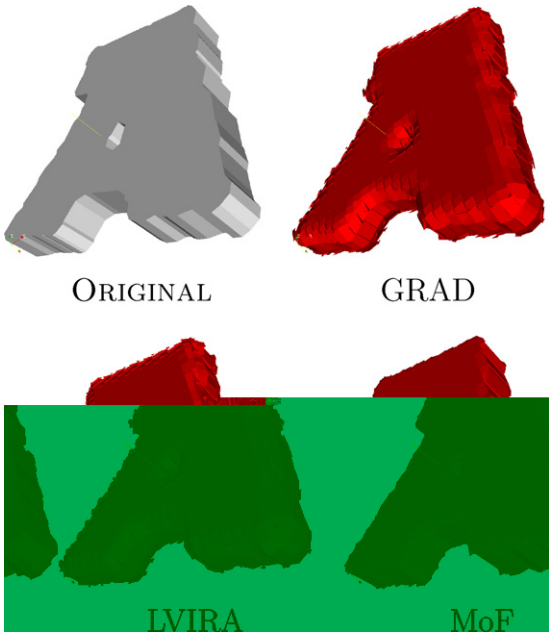

Fig. 15. Interface reconstruction of a not simply connected material region with sharp corners.

4. Multi-material interface reconstruction

An example of a multi-material configuration is illustrated in Fig. 17. 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 are centered at $(0.4, 0.4, 0.4)$ and $(0.6, 0.6, 0.6)$, respectively. Four different materials, shown as green, red, blue and gray regions, are described with two mutually intersecting spherical surfaces.

The configuration presented in Fig. 17 will be used often in this section to compare the different interface reconstruction methods.

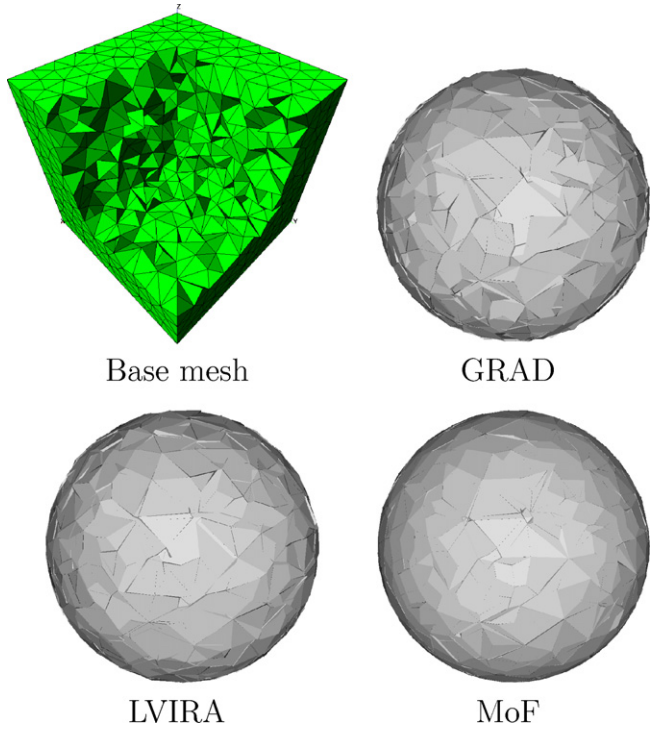

Fig. 16. Two material spherical interface, centered at $(0.5, 0.5, 0.5)$ with radius $r = 0.4$, reconstruction on unstructured tetrahedral mesh ($n_{\text{cells}} = 8659$). Top left – tetrahedral mesh (shown after subcell decomposition) covering the cubic domain of $[0, 1]^3$, Top right – GRAD reconstruction, bottom left – LVIRA reconstruction and bottom right – MoF reconstruction.

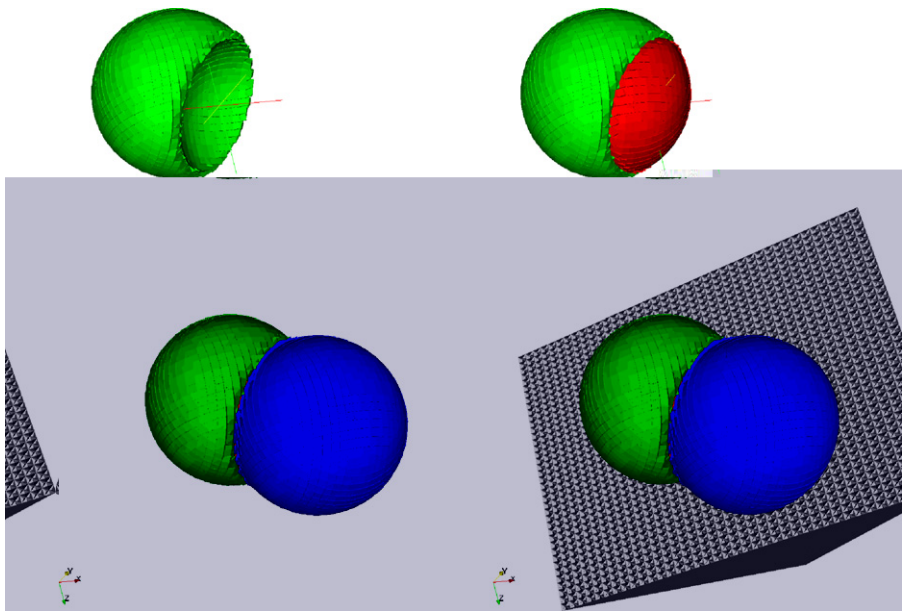

Fig. 17. Example of multi-material configuration. Two intersecting eccentric spheres (S_1 and S_2) define four materials defined with green ($S_1 - S_2$), red ($S_1 \cap S_2$), blue ($S_2 - S_1$), and gray (background) regions. Spheres have same radius of $r = 0.3$, and are centered at $(0.4, 0.4, 0.4)$ and $(0.6, 0.6, 0.6)$, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

To explain the logic of multi-material interface reconstruction, we will first use 2D examples which are easier to visualize. For simplicity of the visualization, we will consider the case when a 2D cell is a convex polygon; therefore, its subdivision into triangles is not shown. In Section 4.1, we will explain the logic of the *nested dissection (ND)* strategy briefly described in Section 1. In Section 4.1.3, we also describe an automatic material ordering in the framework of MoF method. In Section 4.2, we will consider a *group nested dissection (GND)*, where the nested dissection strategy is applied to several groups of materials. The GND strategy allows for exactly reconstructing more complicated multi-material configurations.

Most of this section considers MoF method, where the normal to interface is defined as part of the optimization process. Hence, there is no ambiguity in the normal computation of MoF method when a material ordering is defined. In Section 4.3, the material ordering issues in the GRAD and LVIRA methods are addressed. In Section 4.4, we describe possible definitions of the interface normals for extensions of the GRAD and LVIRA methods to the multi-material case.

A comparison of the MoF, GRAD and LVIRA method on numerical examples is presented in Section 4.5.

4.1. Nested dissection

The general idea of ND strategy is that materials are cut sequentially from a mixed cell in a specified order as illustrated in Fig. 2. The ND strategy is illustrated below with two representative cases of a multi-material configuration: a T-junction case and layered structure case.

4.1.1. T-junction configuration

Let us demonstrate the ND strategy on the example of a T-junction configuration in 2D – Fig. 18.

In ND scheme one needs to specify a material ordering. Let us analyze what will happen if we choose material ordering as: (B, A, C) – which is an optimal ordering for this T-junction configuration. Recall in the Introduction that material #1 in material ordering is material A, material #2 is material B and material #3 is material C.

According to the ND scheme, B is separated from the rest of the materials in the mixed cell. This is done using the MoF method for the two material case, where the first material is B and second material is the union of materials $= A$ and 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 a pure sub-cell Ω_1, as shown in Fig. 19a, which represents material B, and set $\Omega \backslash \Omega_1$ which contains materials A and C. The next (and in case of three materials, the last) step is to create a pure sub-cell which represents material A. This again can be done by applying two material algorithm to $\Omega \backslash \Omega_1$, where the first material will be material A and the second material will be material C. Here, one needs to understand that the second step is performed with respect to set $\Omega \backslash \Omega_1$. Therefore, volume fractions for mate-

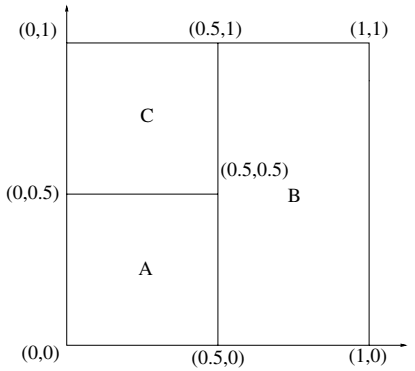

Fig. 18. Three material T-junction configuration. The entire cell Ω is unit square. Materials A and C occupy the left-half of the cell and material B occupies the right-half of the cell. Corresponding volume fractions and centroids are: $f_A = 0.25$, $f_B = 0.5$, $f_C = 0.25$; $\mathbf{x}_c^A = (0.25, 0.25)$, $\mathbf{x}_c^B = (0.75, 0.5)$, $\mathbf{x}_c^C = (0.25, 0.75)$.

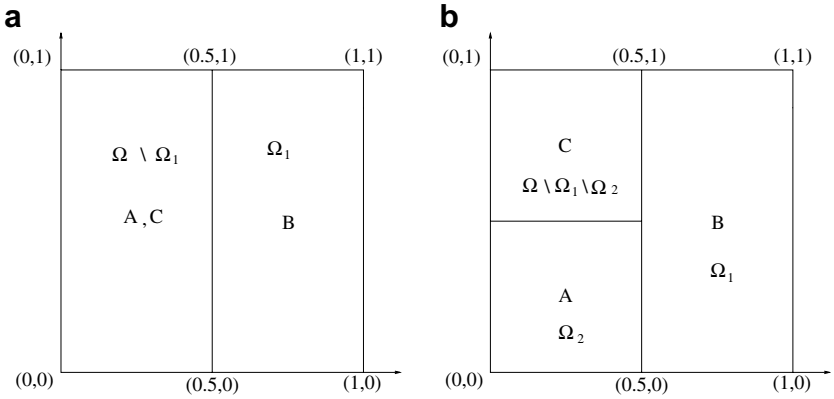

Fig. 19. ND interface reconstruction of T-junction – (B, A, C) ordering: (a) result of the first dissection and (b) result of the second (final in three material case) dissection.

rials A and C have to be recomputed, and in our case both will be equal to 1/2. This second step, as shown in Fig. 19(b), creates a pure sub-cell representing material A, Ω_2. Finally, material C is represented by the pure subcell $\Omega\backslash\Omega_1\backslash\Omega_2$. For material ordering (B, A, C) we will recover the original T-junction material configuration.

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

Now let us consider a different ordering (A, B, C). In this case, analogs of Fig. 19 are presented in Fig. 20 with some additional information about the locations of reference and actual centroids.

In the true configuration (Fig. 18), material A is square; that is, its interface with the other materials is broken line and cannot be exactly represented by one straight line. For this chosen material ordering (A, B, C), after the first dissection we obtain a sub-cell (triangle) Ω_1 (Fig. 20(a)) which represents material A. The area of this triangle equals the original area occupied by material A, but its centroid, marked by $*$, is different from the reference centroid marked by $\times$. The final result of reconstruction is shown in Fig. 20(b). It also shows the positions of actual and reference centroids for the rest of the materials. For this material ordering, the total centroid discrepancy (error) $E_{(A,B,C)}$ in the position of the centroids, defined by

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

is 0.0366461. There are four other material orderings possible, but, because of symmetry of T-junction configuration, only one – (C, B, A) produces the distinct result presented in Fig. 21.

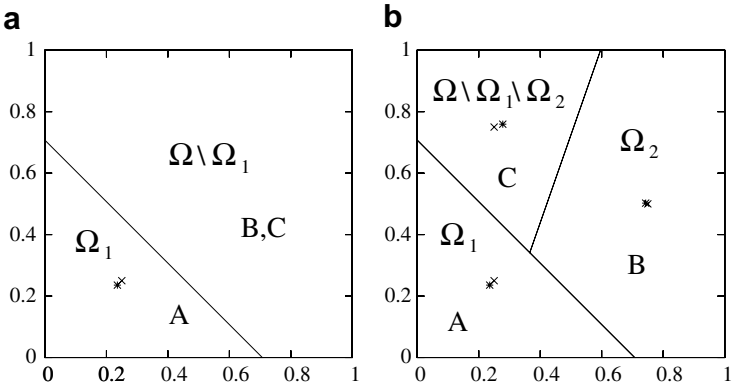

Fig. 20. ND interface reconstruction of T-junction – (A, B, C) ordering: (a) result of first dissection and (b) final result. Reference centroids are marked by $\times$ and actual centroids of reconstructed pure sub-cell are marked by $*$.

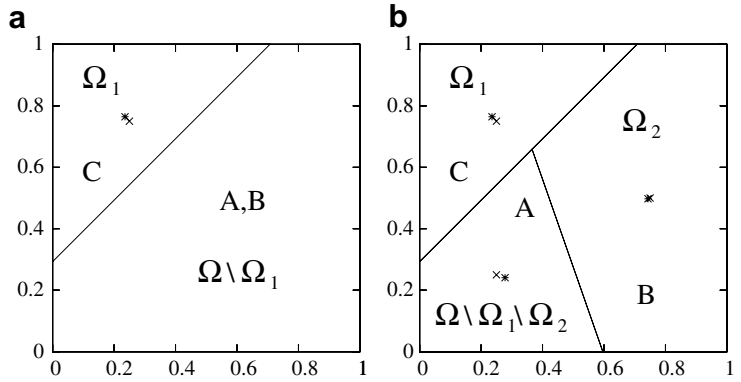

Fig. 21. ND interface reconstruction of T-junction – (C, B, A) ordering: (a) result of the first dissection and (b) result of the second (final) dissection. Reference centroids are indicated by × and actual centroids are marked by ∗.

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

4.1.2. Layered configuration

Here is another configuration where a similar strategy holds. It is the case of "layered" materials as shown in Fig. 22. The entire cell Ω is a unit square. Material A occupies the triangle with vertices: $(0, 0.1)$, $(0.3, 1)$, $(0, 1)$. Material B occupies the pentagon with vertices: $(0, 0.1)$, $(0, 0)$, $(0.5, 0)$, $(0.6, 1)$, $(0.3, 1)$. Finally, material C occupies the quadrangle with vertices: $(0.5, 0)$, $(1, 0)$, $(1, 1)$, $(0.6, 1)$. The corresponding volume fractions and centroids are: $f_A = 0.135$, $f_B = 0.415$, $f_C = 0.45$; $\mathbf{x}_c^A = (0.1, 0.7)$, $\mathbf{x}_c^B = (0.3329, 0.4550)$, $\mathbf{x}_c^C = (0.7740, 0.4814)$.

The original configuration is reproduced exactly with any material ordering where material B is not in the first position. The result of reconstruction for material ordering (B, C, A) presented in Fig. 23(a). In this case, the discrepancy is 0.384038. It is interesting to note that, for this ordering, the reference centroid for material A is located outside the corresponding reconstructed pure sub-cell. This configuration also demonstrates that, in general, the results of reconstruction when two last materials are interchanged are not the same. The result of reconstruction for material ordering (B, A, C) is presented in Fig. 23(b). In this case, the discrepancy is 0.342561. Clearly in the case of layered materials, we can find the optimal ordering by just comparing how well the actual centroid of first material matches the corresponding reference centroid.

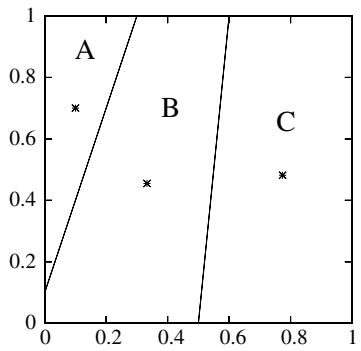

Fig. 22. Three material *layered* configuration. Positions of reference centroids are marked by × and actual centroids by ∗. The original configuration is reconstructed exactly with material orderings (A, B, C), (A, C, B), (C, B, A), or (C, A, B).

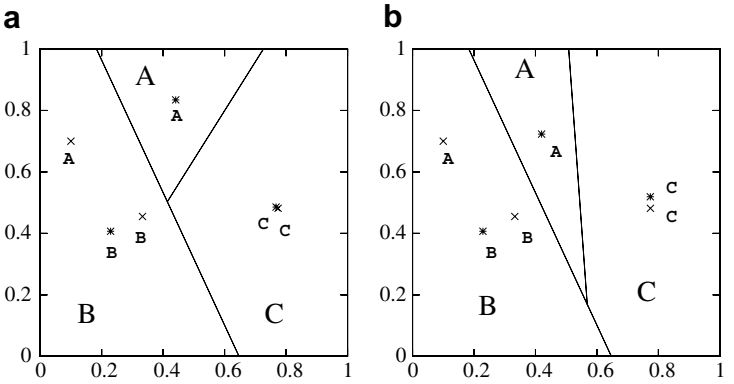

Fig. 23. ND interface reconstruction of layers. (a) Material ordering (B, C, A), reference (×) and actual (∗) centroids are also shown. These centroids are also marked by their corresponding material index. (b) Material ordering (B, A, C).

4.1.3. Automatic ordering for ND

In search of the best material ordering for nested dissection, two issues have to be addressed. First issue 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 would have to change from cell to cell, i.e. *local* ordering. The next issue is how to find the optimal order. The material ordering may be assigned by the user, but ideally the optimal order would be determined without user intervention, i.e. *automatic* ordering. Due to their non-local nature, it is not clear how to extend the GRAD and LVIRA reconstruction schemes to the multi-material case with automatic-local ordering. However, the MoF method, with its local nature, can be applied with the automatic-local ordering.

The automatic ordering determination strategy is presented below with our primary reconstruction scheme, MoF. The algorithm as presented in [4] is a straight forward extension from two material interface reconstruction. In the two material case, the interface is determined such that the deviation of an actual centroid from the reference is minimized. The same strategy can be applied for multi-material interface reconstruction as well as for proper ordering selection. For any given material order, $\mathbf{m}$, the total deviation of actual centroids from the reference centroids can be computed as

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

where $\mathbf{m} = \left(m_1^{\mathrm{id}}, m_2^{\mathrm{id}}, \ldots, m_n^{\mathrm{id}}\right)$ is an ordering vector containing material-ids, $\mathbf{N} = (\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_{n-1})$ is an ordered list of corresponding interface normals computed by MoF optimization, n is the total number of materials inside the cell, $\mathbf{x}_c^{\mathrm{ref}}\left(m_i^{\mathrm{id}}\right)$ is the reference centroid of material m_i^{id} and $\mathbf{x}_c^{\mathrm{act}}\left(m_i^{\mathrm{id}}, \mathbf{n}_i\right)$ is the actual centroid of material m_i^{id} reconstructed with interface normal $\mathbf{n}_i$. For cells with n materials, the number of all possible orderings ($\mathbf{m}$) is $n!$. The optimal ordering, $\mathbf{m}^*$, represents the least accumulated error in centroids as expressed in Eq. (13).

This way of searching optimal orderings results in an exhaustive test, i.e. $n!$ evaluations of Eq. (13). This, however, leaves the door open to possible short cuts. For example, part of the ordering vector can be fixed for materials of which the centroid error of reconstruction is below a certain tolerance. In this paper, the exhaustive search is performed for cells containing more than two materials.

4.2. Group nested dissection

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

4.2.1. Motivation

Let us consider a "four corner" four material configuration (+ junction) shown in Fig. 24. The results of applying the ND scheme for two possible material orderings is presented in Fig. 25.

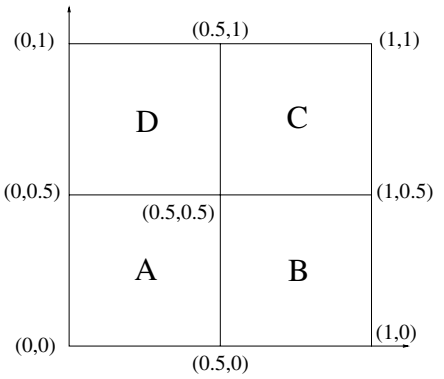

Fig. 24. Four material – "four corner" (+ junction) configuration. The entire cell Ω is the unit square. Each material occupies one quadrant of the cell. Corresponding volume fractions and centroids are: $f_A = f_B = f_C = f_D = 0.25$; $\mathbf{x}_c^A = (0.25, 0.25)$, $\mathbf{x}_c^B = (0.75, 0.5)$, $\mathbf{x}_c^C = (0.5, 0.75)$, $\mathbf{x}_c^D = (0.75, 0.75)$.

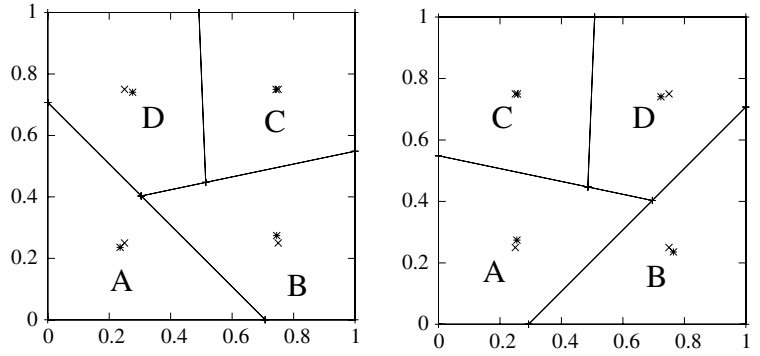

Fig. 25. "Four corner" configuration. Interface reconstruction using nested dissection scheme: left – (A, B, C, D) ordering and right – (B, A, C, D) ordering. Positions of reference, $\times$, and actual centroids, $*$, are also shown.

From Fig. 25, it is clear that there is no material ordering in the framework of ND scheme which will allow exact reconstruction of the "four corner" (+ junction) configuration. The reason for this is that, there is no material whose interface can be represented with a straight line.

4.2.2. GND

Let us revisit the ND scheme. Two main components of this scheme is (1) on each step we use the algorithm for two materials and (2) each time we dissect the cell. It is clear that instead of dissecting one from the rest, we can dissect any *group* of materials from the rest. In case of "four corner" configuration one can choose the first group $G_1 = [A, D]$ and second group $G_2 = [B, C]$. Here, the volume fractions are $f_{G_1} = f_{G_2} = 0.5$ and the centroids are $\mathbf{x}^{G_1} = (0.25, 0.5)$, $\mathbf{x}^{G_2} = (0.75, 0.5)$. 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 $G_{2,1} = B$, $G_{2,2} = C$, and apply GND (which will now be the same as ND) for each group dissecting them into corresponding subgroups. This algorithm reproduces the "four corner" (+ junction) configuration exactly.

Let us note that the ND strategy with material ordering (A, B, C, D) can be considered as GND with $G_1 = A$, $G_2 = [B, C, D]$, $G_{2,1} = B$, $G_{2,2} = [C, D]$ and finally $G_{2,2,1} = C$, $G_{2,2,2} = D$.

In general, in the GND scheme, the role of the single material in the ND scheme is played by a group of materials, and then in each group, the GND scheme is applied again until sub-groups consist of one material. For example, for the configuration presented in Fig. 26, the GND scheme is: $G_1 = [A, E]$, $(G_{1,1} = A, G_{1,2} = E)$; $G_2 = [B, C, D]$, $(G_{2,1} = B, G_{2,2} = [C, D], (G_{2,2,1} = C, G_{2,2,2} = D))$.

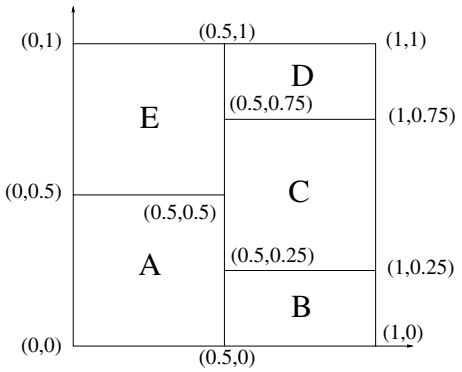

Fig. 26. Five materials – "double T-junction" configuration. Entire cell Ω is the unit square. Corresponding volume fractions and centroids are: $f_A = 0.25$, $f_B = 0.125$, $f_C = 0.25$, $f_D = 0.125$, $f_E = 0.25$; $\mathbf{x}_c^A = (0.25, 0.25)$, $\mathbf{x}_c^B = (0.75, 0.125)$, $\mathbf{x}_c^C = (0.75, 0.5)$, $\mathbf{x}_c^D = (0.75, 0.875)$, $\mathbf{x}_c^E = (0.25, 0.75)$.

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

4.2.3. Automatic ordering for GND

In this section, we give a more formal description of ideas described in the previous section. If a cell contains four or more materials, the possibility of interface formation is more complex. For examples shown in Fig. 27, a cell with a quadruple point or multiple triple points, it can produce a group structure inside of a cell. In such cases, the ND strategy fails to recover the correct interfaces, so GND should be considered as a possible candidate of optimal ordering of cells with $n_{\text{mat}} \geqslant 4$.

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

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

where the grouping vector $\mathbf{g} = (g_1, g_2, \ldots, g_{n-1})$ indicates the cutting position within a given material order $\mathbf{m} = \left(m_1^{\text{id}}, m_2^{\text{id}}, \ldots, m_n^{\text{id}}\right)$. Each component g_i of the grouping vector uniquely indicates the cutting position (e.g. if $g_1 = 2$, then the first cut is after the second material in the ordering vector) within the given ordering

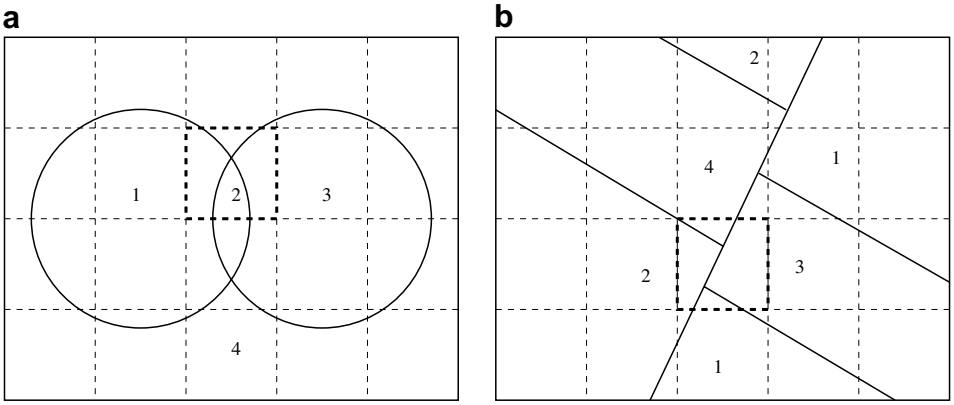

Fig. 27. Interface structures of grouped multi-materials in 2D. (a) A quadruple point and (b) multiple triple points in a cell (indicated with bold dashed lines). Multiple triple points (b) can be removed by mesh refinement, but the quadruple point (a) persists regardless of mesh resolution.

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

In the four material case, the number of possible sequential material orderings is 4!. For instance, with $\mathbf{m} = (2, 3, 1, 4)$, there is only one possible grouping because the result of grouping $\mathbf{g} = (2, 1, 3)$,

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

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

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

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

As the number of materials grows ($n_{\text{mat}} \geqslant 6$), exhaustive search can also express multi-layered grouping, i.e. groups of groups. For example, in a cell containing seven materials as shown in Fig. 28, a multi-layered grouping described by $\mathbf{m} = (1, 2, 3, 4, 5, 6, 7)$ and $\mathbf{g} = (3, 1, 2, 5, 4, 6)$ 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. 28. After all possible combinations ($n!(n-1)!$ 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 that for the two material case. In the *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, the second material from the rest, and so on. In *group nested* dissection as illustrated in Fig. 29, interfaces are reconstructed by separating each material group from the rest, until all groups are composed of a single material. Hence, nested dissection can be considered as a special case of group nested dissection with all uni-material groups.

The difference in the intersection algorithm, when 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

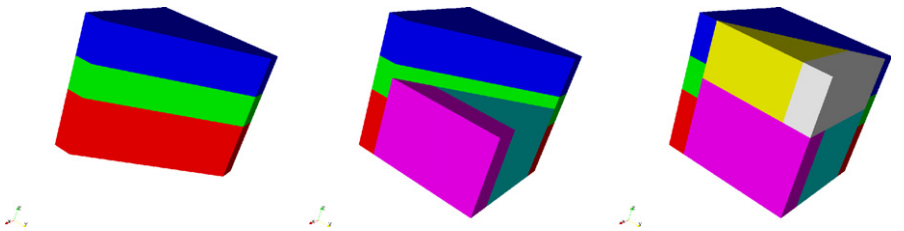

Fig. 28. Multi-layered material grouping with a cell containing seven materials. From the left, groups of materials are added in the order of: the first group of three materials (red, green and blue), the first sub-group of the second group (cyan and magenta) and the second sub-group of the second group (yellow and gray). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

example, a grouped intersection of n materials $(1,2,\ldots,n)$ will needs 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 the centroid of material-i, and $w_i = f_i/f_G$ is the weight for material-i.

4.3. Material ordering for GRAD and LVIRA

As we already mentioned, for the MoF method, the optimal material grouping and group ordering can be determined by first trying all possible ordering and groups and then choosing one which has smallest 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, the MoF method is applied to each mixed cell independently and therefore the material ordering is inherently local, that is it can be different for each mixed cell.

The choice of material ordering for traditional PLIC methods is unsolved problem. There have been several attempts to develop algorithms to automate material ordering [51,47,38,43]. All of them are based on using information about exact or approximate centroid positions of the materials in a mixed cell and its neighbors. To best of our knowledge, (see, e.g. examples in [4], and also comments in [47,38]

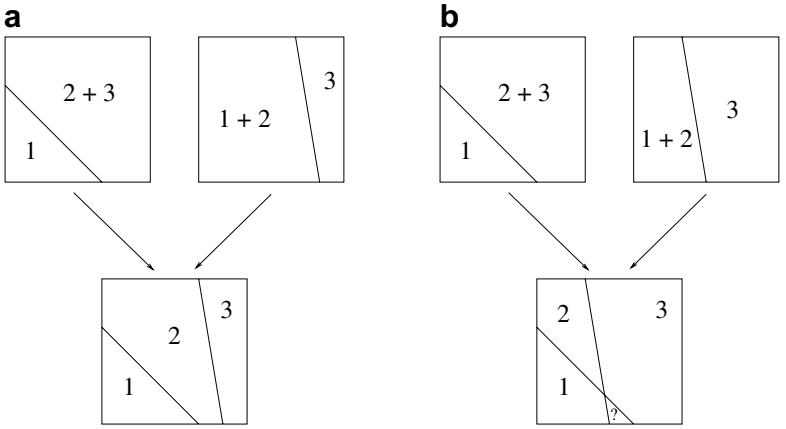

Fig. 30. Onion skin interface reconstruction: (a) valid interface reconstruction and (b) invalid interface reconstruction with intersecting interfaces.

determined by techniques developed for two material configuration. The onion skin model is demonstrated in Fig. 30 for case of three materials.

The onion skin model was originally developed for layered materials. The left picture in Fig. 30 presents a valid interface reconstruction for case of layered materials. The picture on the right in Fig. 30 presents a situation when interfaces constructed using the onion skin model intersect each other. This problem is well known [31,52,44]. For our purposes, an interface reconstruction method which does not always produce pure sub-cells for each material is not unacceptable.

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

According to the ND scheme, which is illustrated in Fig. 2, we use the normal between material #1 and the rest of the materials (computed as in onion skin model) to construct a pure sub-cell containing this material (Ω_1 in Fig. 2). Now, domain $\Omega\backslash\Omega_1$ contains materials $\#2, 3, \ldots, n$. To create a pure sub-cell for material #2, we use the onion skin normal computed for mixture of materials #1 and #2 and mixture of the materials $\#3, 4, \ldots, n$. This normal is used to cut a pure sub-cell Ω_2 from $\Omega\backslash\Omega_1$. This process is repeated until pure sub-cells are constructed for all materials.

The difference between the onion skin interface reconstruction and the ND interface reconstruction with onion skin normals is shown in Fig. 31 for the case of three materials.

In the left picture of Fig. 31, we repeat the result of onion skin interface reconstruction from Fig. 30(b), where interfaces are intersect each other. In the right picture of Fig. 31, which corresponds to the ND scheme with the same set of normals, the position of the interface computed using the second normal is adjusted (moved to the left) such that material #2 has the specified volume.

Let us note again that onion skin normals are always computed for two "materials", each being a mixture of original materials according to the chosen material ordering. Therefore, for GRAD and LVIRA, the com-

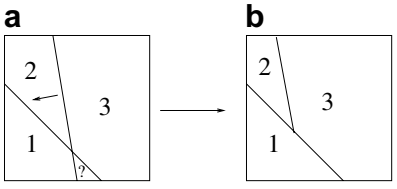

Fig. 31. Comparison of onion skin interface reconstruction – left and ND interface reconstruction with onion skin normals.

putation of these normals will be done using the corresponding algorithms for two materials as described in Section 3.

For the GND scheme, each group is considered as a material for purpose of computing of the corresponding normal.

4.5. Numerical examples

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

4.5.1. Mesh convergence study

A mesh convergence study is presented for a multi-material configuration. The three-material configuration, shown in Fig. 32, is utilized. For visualization purposes, only the fraction of the background (gray), which originally surrounds the other materials (cyan and magenta), is displayed. To exclude any possible effect of ordering, all three PLIC methods presented are applied with the same prescribed global ordering (cyan → magenta → gray). The global ordering is chosen such that the triple lines (where the three materials merge) are resolved best for all three methods. This ordering can be also chosen by selecting the best order to paint the given material configuration in 2D.

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

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

where $\mathbf{m}_{\text{interior}} = \{1,2,\ldots,n_{\text{mat}}-1\}$ is set of all materials except the background (for the example in Fig. 32, $\mathbf{m}_{\text{interior}} = \{1,2\}$), and T and R are sets of true and reconstructed material regions.

The actual reconstruction for the multi-material configuration is shown in Fig. 33. MoF shows the best reconstruction for all range of meshes. LVIRA shows the second most accurate result on fine meshes. GRAD results are better than LVIRA for low resolution, but is the least accurate as the mesh is refined. Along the triple lines (sharp edges of the concave sphere where the two spheres and background are merging), only MoF reconstructs them with sharp angles, but LVIRA and GRAD smears the angle near the triple lines.

The rate of convergence is displayed on Fig. 34. The overall behavior is in accordance with the result of the two material case as shown in Fig. 12. In general, MoF is the most accurate method with second-order accuracy, followed by another second-order accurate method, LVIRA and GRAD is the least accurate, being of first-order.

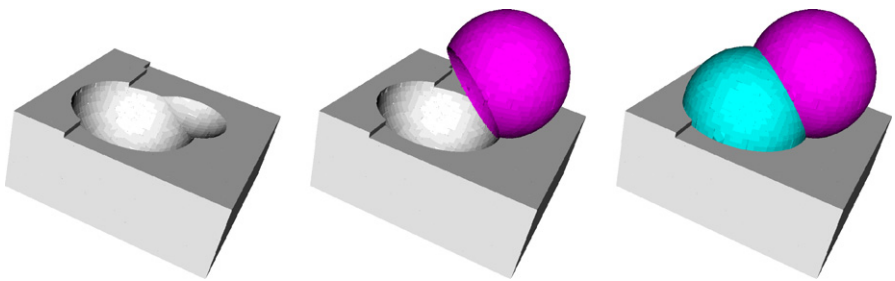

Fig. 32. Multi-material configuration for the mesh convergence study ($n_{\text{mat}} = 3$). The configuration is similar to the one shown in Fig. 17. Two intersecting eccentric spheres (S_1 and S_2, centered at $(0.4,0.4,0.4)$ and $(0.6,0.6,0.6)$, respectively, both with radius $r = 0.3$) define three materials; #1 defined by S_1 (cyan), #2 by $S_2 - S_1$ (magenta) and #3 by the background (gray). From the left, only a fraction of #3, and all of #2 and #1 are displayed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

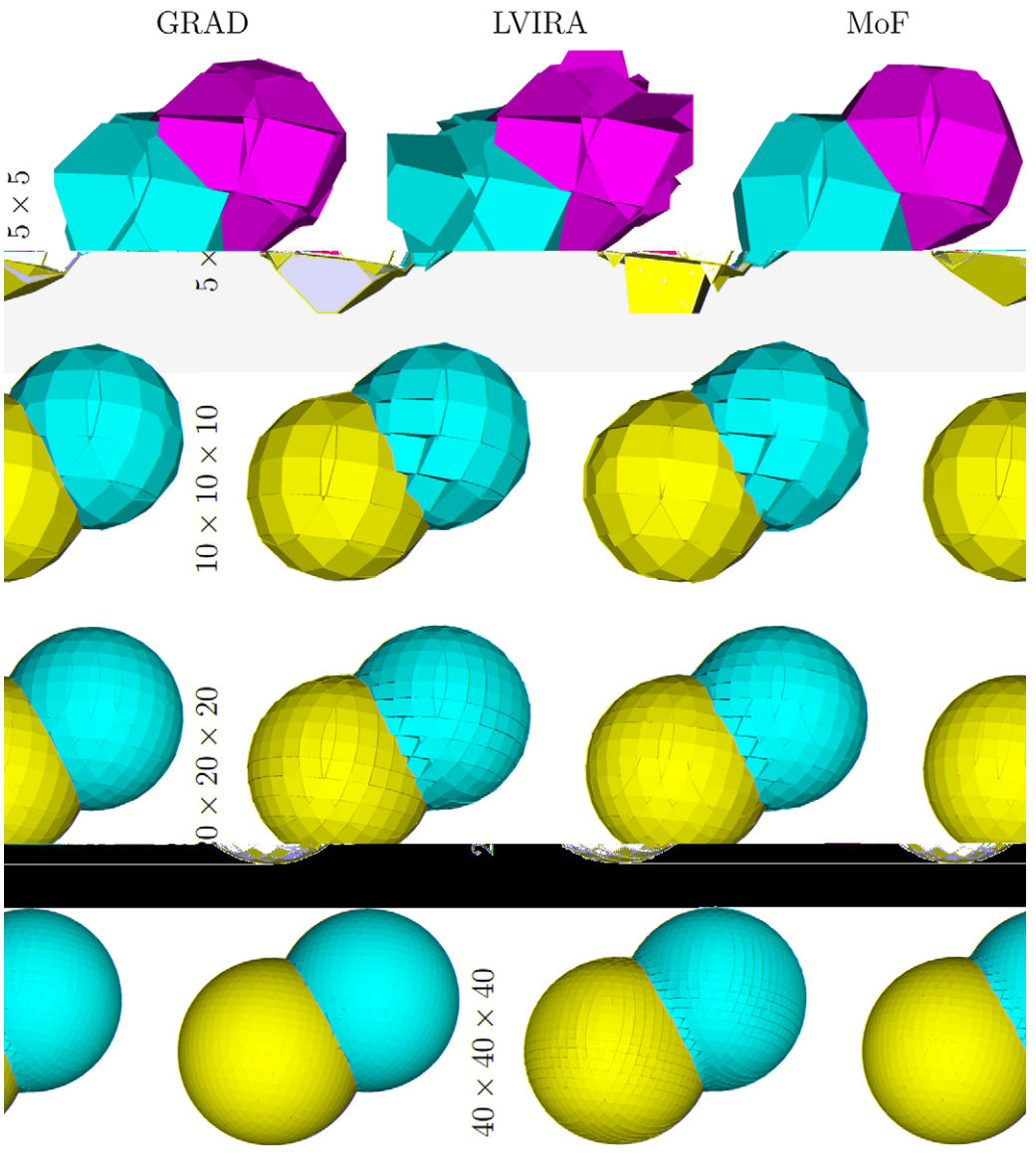

Fig. 33. Mesh convergence study for the multi-material case ($n_{mat} = 3$). Global ordering of (#1, #2, #3) are used for all reconstruction methods. For GRAD and LVIRA, interface normals are pre-computed by the onion skin strategy, then the ND strategy is applied with given global ordering. Equispaced hexahedral meshes covering the domain of $[0,1]^3$ are used for interface reconstruction.

4.5.1.1. Computational cost. The overall computational cost of the reconstruction methods are measured in CPU time in Table 1. The machine is an 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 reports only a rough estimate for each method.

Regardless of the mesh resolution, GRAD is the fastest method and LVIRA is the most expensive method. At the finest level of the mesh ($40 \times 40 \times 40$), MoF takes 6.7 times of CPU time compared to that of GRAD and LVIRA takes 9.1 times of CPU time compared to the GRAD. The overall behavior of CPU cost with respect to the mesh size is displayed in Fig. 35. It is clear that all three methods show linear complexity with respect to the mesh resolution. At each step of mesh refinement, the number of mixed cells approximately quadruples and the CPU time is of O(n) complexity, where n is number of mixed cells.

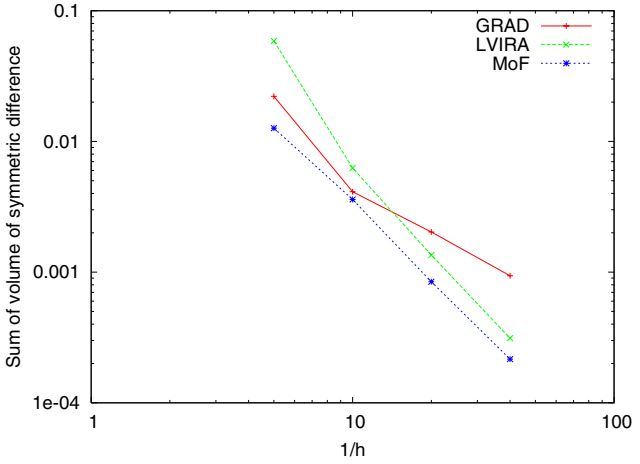

Fig. 34. Rate of convergence for multi-material interface reconstruction. The convergence rates for the last refinement step are: $p^{\mathrm{GRAD}} = 1.169$, $p^{\mathrm{LVIRA}} = 2.034$, $p^{\mathrm{MoF}} = 1.966$.

Table 1
CPU time required for reconstruction of the three-material configuration shown in Fig. 33

Mesh resolution	# $\mathrm{cells}_{\mathrm{mixed}}$	GRAD	LVIRA	MoF
$10 \times 10 \times 10$	222	4^s	52^s	36^s
$20 \times 20 \times 20$	1110	24^s	325^s	188^s
$40 \times 40 \times 40$	4614	148^s	1347^s	996^s

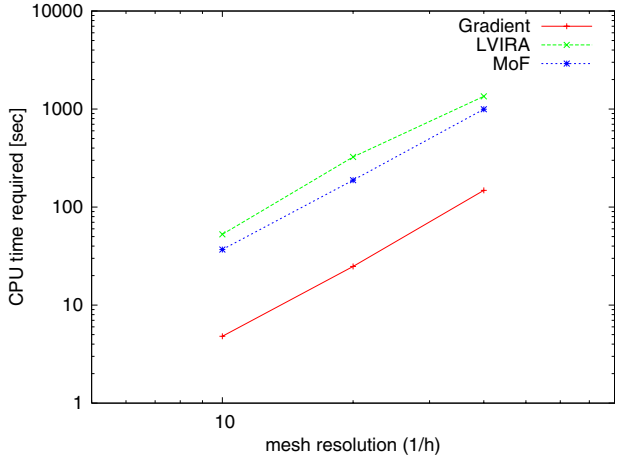

Fig. 35. CPU time required with respect to mesh resolution for the reconstruction of the three-material configuration displayed in Fig. 33. All three methods show linear complexity (O(n) complexity, where n is number of mixed cells) with respect to the mesh resolution. At each step of successive refinement, the number of mixed cells and the CPU time quadruples.

For large scale calculations, parallelization is straightforward in MoF by the virtue of its locality, i.e. no intercell communication is necessary for interface reconstruction.

4.5.2. Examples of nested dissection

In this section, the effectiveness of the automatic-local ordering strategy together with the ND strategy (Section 4.1) is demonstrated with our primary reconstruction scheme, MoF. For GRAD and LVIRA, the *onion*

skin ND strategy is employed as explained in Section 4.4. In onion skin ND, first the interface normals are pre-computed by conventional onion skin model, and then the ND strategy is applied with a pre-determined ordering given by MoF.

4.5.2.1. 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. 36. Carefully selected global ordering can

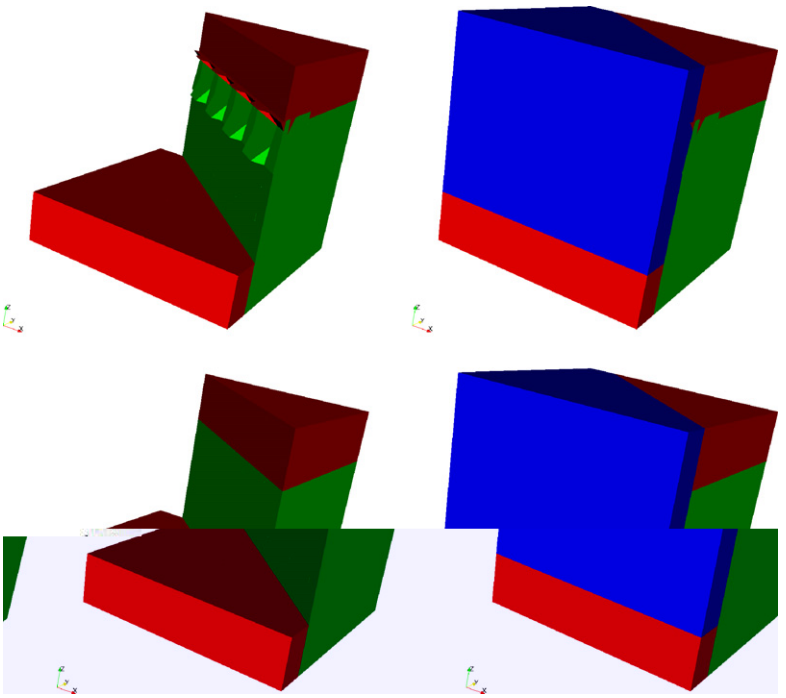

Fig. 36. Global vs. local ordering for MoF reconstruction of interfaces with multiple triple lines on a 5^3 hexahedral mesh. Global ordering (top) cannot resolve all triple lines but the local ordering (bottom) can resolve both due to its adaptation capability in ordering.

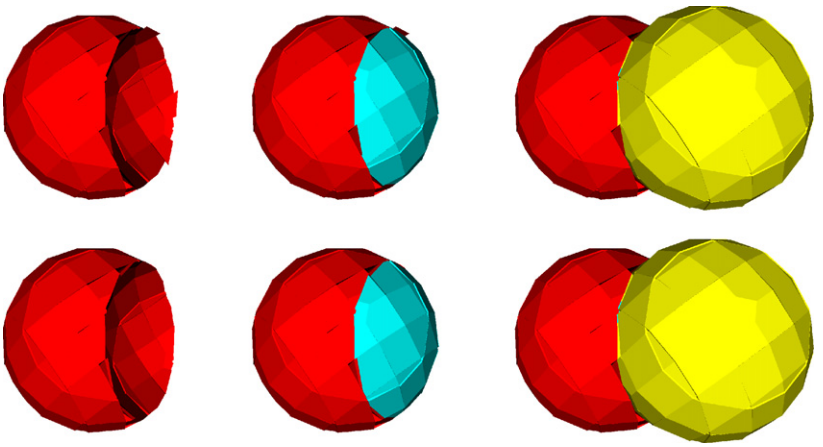

Fig. 37. Global vs. local ordering for MoF reconstruction of two sphere configuration (as illustrated in Fig. 17) on 10^3 hexahedral mesh. Global ordering (top) cannot resolve clean intersecting edges (especially clear on left and middle columns of pictures), but the local ordering (bottom) resolves the clean edges with adaptive ordering from cell to cell. Each material (except the background) is displayed progressively from left-to-right.

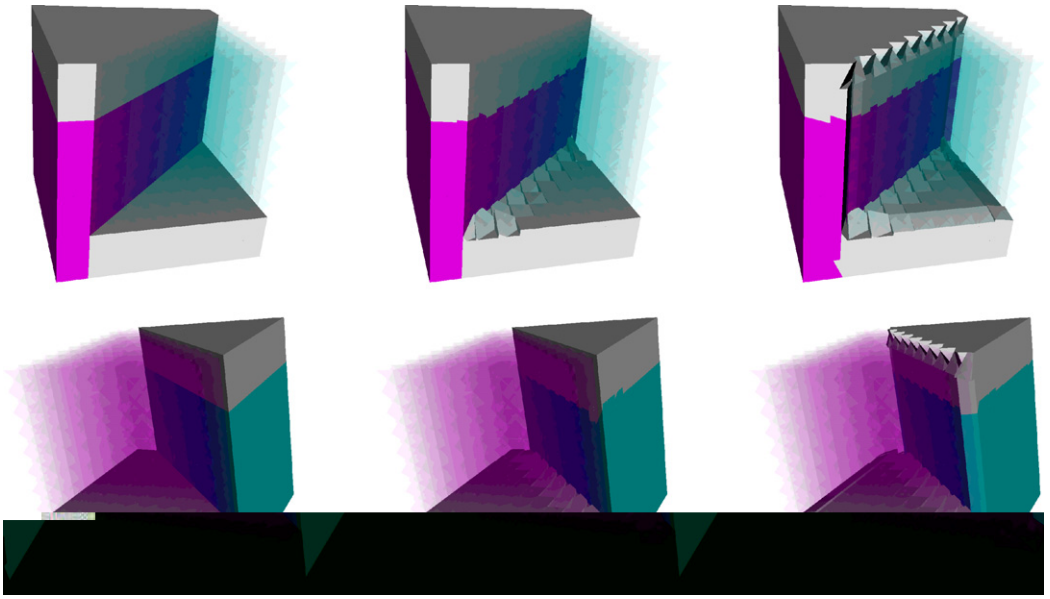

Fig. 38. MoF, LVIRA and GRAD (from the left) interface reconstruction of double triple lines with the pre-determined local ordering from MoF. Double triple lines are reconstructed correctly by MoF with local ordering. However, *onion skin* reconstructions of GRAD and LVIRA fail to recover the double triple lines even with the same local ordering as MoF. For visualization, the top row is inverted upside-down and left-to-right at the bottom with alternating transparent region. A 10^3 hexahedral mesh is used.

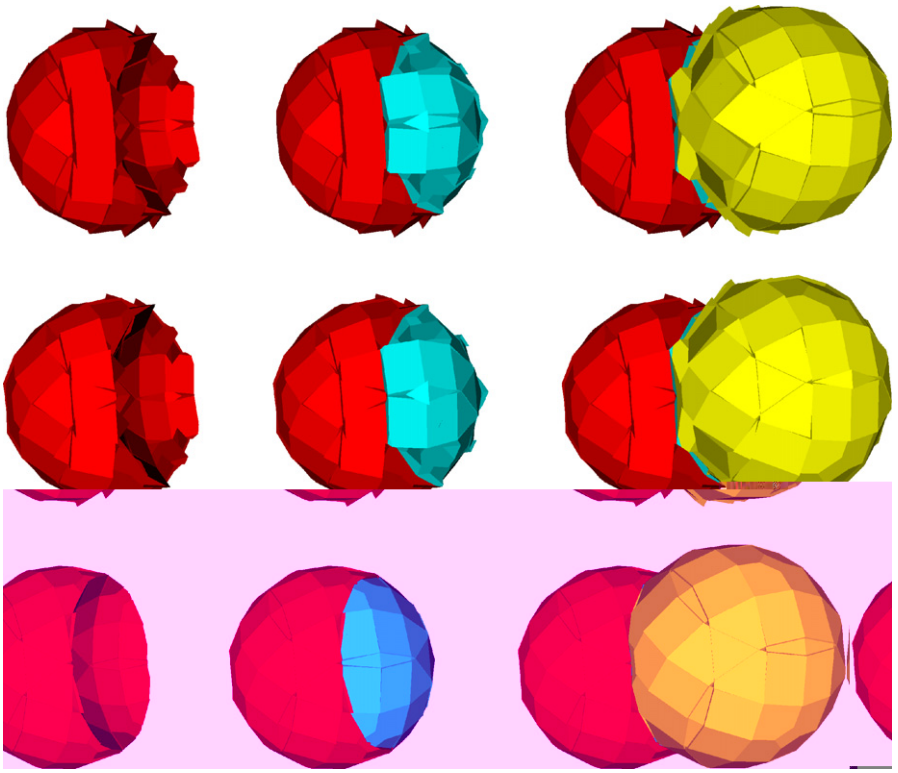

Fig. 39. GRAD, LVIRA and MoF (from the top) interface reconstruction of the two sphere example with a pre-determined local ordering given by MoF. MoF shows the best recovery of the two intersecting spheres with automatic-local ordering, but the *onion skin* normals used in GRAD and LVIRA reconstruction methods poorly recover the intersecting spheres even with the same local ordering as MoF. A 10^3 hexahedral mesh is used.

recover a single triple line, but not all of them. 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 in Fig. 37. 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 the two spheres, the global ordering shows a rough reconstruction of red material. On the contrary, the automatic-local ordering shows clean edge reconstruction, which confirms the superiority of automatic-local ordering.

4.5.2.2. Comparison of MoF with LVIRA and GRAD. In contrast to the straightforward extension of MoF to multi-material automatic-local ordering, LVIRA and GRAD have an inherent restriction on multi-material generalization due to its non-locality. Hence, for the comparison purpose of the methods in the multi-material case, the local orders obtained by MoF are utilized for the *onion skin* ND interface reconstruction of LVIRA and GRAD.

In Fig. 38, the three methods are compared with a 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 fail to recover either one.

Fig. 39 emphasizes the superiority of MoF over other methods in the 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 examples emphasize the effectiveness of automatic-local ordering in MoF for multi-material interface reconstruction.

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

4.5.3. Examples of group nested dissection

The effectiveness of grouping in multi-material interface reconstruction is illustrated in Fig. 40. The computational domain containing four materials is discretized into 3^3 hexahedral cells. In the cells containing all four materials, nested dissection barely recovers a single triple line, but grouping shows perfect recovery of both triple lines. In this particular interface configuration, mesh refinement will not produce cells with two triple lines.

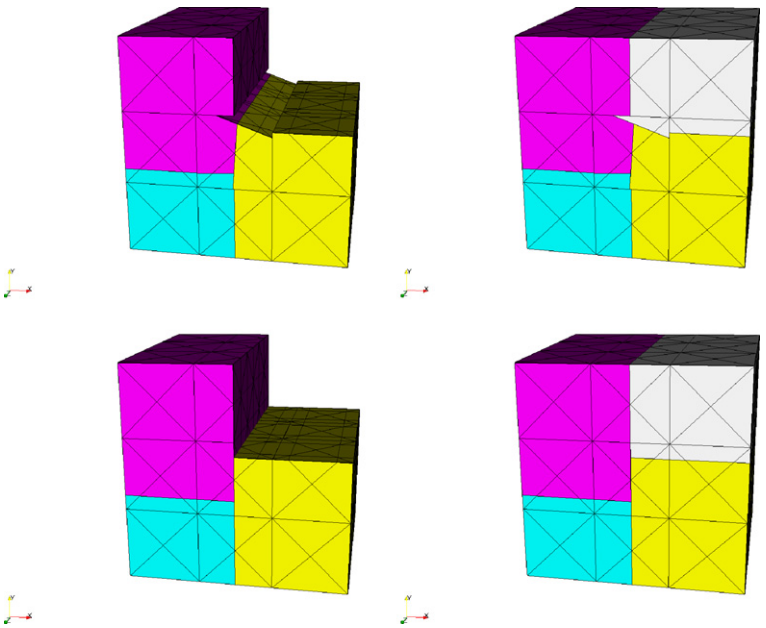

Fig. 40. ND vs. GND. Nested dissection (top) cannot recover dual triple lines at the central cell, but the group nested dissection (bottom) recovers the triple lines correctly. A 3^3 hexahedral mesh is used.

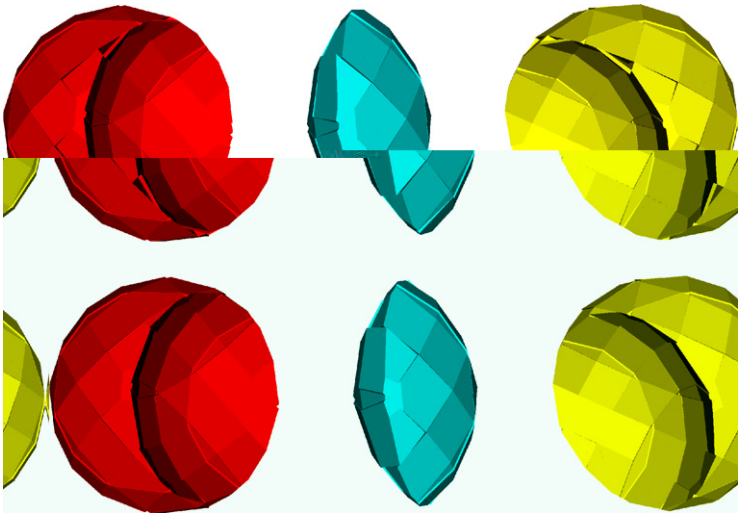

Fig. 41. ND vs. GND with two sphere example. Nested dissection (top) shows dimples around the edges (quadruple lines) where four materials merge, but the group nested dissection recovers the interface without dimples. Each material region is displayed in different perspectives. A 10^3 hexahedral mesh is used.

The last example of GND is displayed in Fig. 41. 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 refinement cannot be a remedy because there will be always cells with quadruple lines. The effectiveness of grouping is clear in Fig. 41. Nested dissection reveals a discontinuity on the spherical surface near the quadruple lines, but the grouping removes such discontinuity, which emphasizes the advantage of GND.

4.5.4. Example of complex geometry

In this section, the effectiveness of the MoF method is demonstrated with complex geometries. The original material regions for reconstruction are represented with tetrahedral volume meshes as illustrated in Fig. 43. The initial geometry of *bolt* & *nut* is obtained from a CUBIT [63] surface mesh, and the volume meshes are generated by the aft3D library [64] based on the surface representation. The initialization of volume fraction and centroid data is carried on the base mesh as shown in Fig. 42 by exact intersection between *bolt* & *nut*

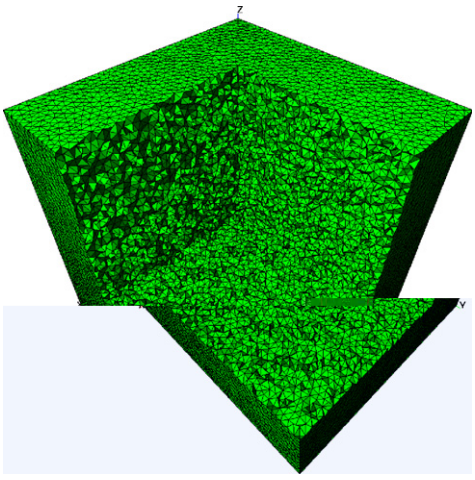

Fig. 42. Unstructured tetrahedral base mesh ($n_{\text{cells}} = 555,468$) used for reconstruction of *bolt* & *nut* geometry as shown in Fig. 43. The base mesh is generated by Gmsh [61].

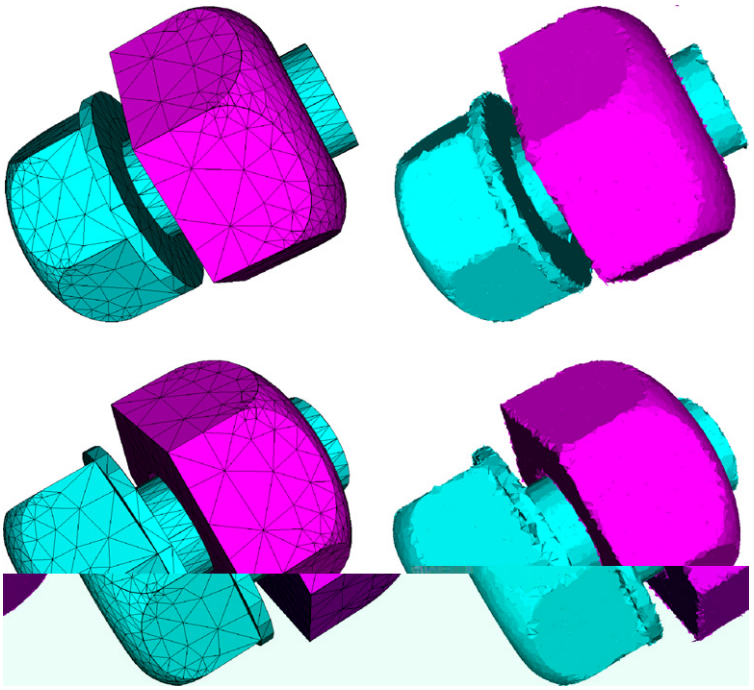

Fig. 43. Multi-material ($n_{\text{mat}} = 3$, i.e. bolt, nut and background) interface reconstruction with MoF. The left column represents original material region represented by tetrahedral volume meshes and the right column shows its reconstruction with the MoF method on the base mesh displayed in Fig. 42. Top and bottom rows show views from different perspectives.

volume mesh and base mesh. The actual reconstruction of the original object is illustrated in Fig. 43. The MoF reconstruction resolves flat as well as curved regions, which demonstrates the effectiveness of MoF method.

5. Conclusion and future work

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

The performance of the methods is demonstrated on numerical examples that involve different types of meshes containing smooth and non-smooth material shapes. A convergence study is performed for the case

of two materials as well as for the multi-material case. The relative computational cost of these methods is also presented.

Numerical examples show that MoF is most accurate method over all levels of the mesh resolution. The ability of the MoF method to automatically choose an optimal local ordering of the materials, the locality of the method (which allows efficient parallel implementation) as well as its accuracy makes MoF our suggested method of choice.

For further work, we will describe how to incorporate the moment of fluid method into the ALE framework. That is we will describe how to obtain information about reference volume fractions and reference centroids and how to perform a multi-material remap including remapping of the centroid positions. Acceleration algorithms for avoiding $n!$ ordering search and the reconstruction sensitivity study on imprecise reference centroids will also be investigated.

Acknowledgments

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, Y. Vasilevsky, A. Barlow, S. Owen and K. Pendley for helpful discussion and valuable comments. The authors also thank B. Wendroff and J. Fung for reading the manuscript and giving valuable comments. This work was supported by the Advanced Simulation and Computing (ASC) program at the Los Alamos National Laboratory.

References

[1] E. Puckett, A volume-of-fluid interface tracking algorithm with applications to computing shock wave refraction, in: H. Dwyer (Ed.), Proceedings of the Fourth International Symposium on Computational Fluid Dynamics, 1991, pp. 933–938.
[2] J.E. Pilliod, E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, Journal of Computational Physics 199 (2004) 465–502.
[3] V. Dyadechko, M. Shashkov, Moment-of-fluid interface reconstruction, Tech. Rep. LA-UR-05-7571, Los Alamos National Laboratory, 2005. Also available as http://cnls.lanl.gov/~shashkov/.
[4] V. Dyadechko, M. Shashkov, Multi-material interface reconstruction from the moment data, Tech. Rep. LA-UR-06-5846, Los Alamos National Laboratory, 2006. Also available as http://cnls.lanl.gov/~shashkov/.
[5] C. Hirt, A. Amsden, J. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, Journal of Computational Physics 14 (1974) 227–253 (reprinted in 135 (1997) 203–216).
[6] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Computer Methods in Applied Mechanics and Engineering 99 (1992) 235–394.
[7] L.G. Margolin, Introduction to "an arbitrary Lagrangian–Eulerian computing method for all flow speeds", Journal of Computational Physics 135 (1997) 198–202.
[8] P. Kjellgren, J. Hyvarien, An arbitrary Lagrangian–Eulerian finite element method, Computational Mechanics 21 (1998) 81–90.
[9] J.S. Peery, D.E. Carroll, Multi-material ale methods in unstructured grids, Computer Methods in Applied Mechanics and Engineering 187 (2000) 591–619.
[10] H.U. Mair, Review: hydrocodes for structural response to underwater explosions, Shock and Vibration 6 (1999) 81–96.
[11] H.T. Ahn, Y. Kallinderis, Strongly coupled flow/structure interactions with a geometrically conservative ALE scheme on general hybrid meshes, Journal of Computational Physics 219 (2006) 671–696.
[12] J. Donea, S. Giuliani, J.P. Halleux, An arbitrary Lagrangian–Eulerian finite element method for transient dynamic fluid–structure interactions, Computer Methods in Applied Mechanics and Engineering 33 (1982) 689–723.
[13] D.E. Burton, Multidimensional discretization of conservation laws for unstructured polyhedral grids, Tech. Rep. UCRL-JC-118306, Lawrence Livermore National Laboratory, 1994.
[14] D.E. Burton, Consistent finite-volume discretization of hydrodynamics conservation laws for unstructured grids, Tech. Rep. UCRL-JC-118788, Lawrence Livermore National Laboratory, 1994.
[15] Y. Kallinderis, H.T. Ahn, Incompressible Navier–Stokes method with general hybrid meshes, Journal of Computational Physics 210 (2005) 75–108.
[16] H.T. Ahn, G.F. Carey, An enhanced polygonal finite-volume method for unstructured hybrid meshes, International Journal for Numerical Methods in Fluids 54 (2007) 29–46.
[17] A.J. Barlow, A new Lagrangian scheme for multi-material cells, in: Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering – ECCOMAS Computational Fluid Dynamics Conference 2001, Swansea, Wales, UK, September 4–7, 2001.

[18] Y. Bondarenko, Y. Yanilkin, Computation of the thermodynamic parameters in the mixed cells in gas dynamics, Mathematical Modeling 14 (2002) 63–81.
[19] R.E. Tipton, Cale mixed zone pressure relaxation model, private communication, 1989.
[20] V.I. Delov, V.V. Sadchikov, Comparison of several models for computation of thermodynamical parameters for heterogeneous Lagrangian cells, VANT, Mathematical Modeling of Physical Processes 1 (2005) 57–70.
[21] B. Despres, F. Lagoutiere, Numerical resolution of a two-component compressible fluid model with interfaces and mixing zones (preprint R04008 of the Laboratoire Jacques-Louis Lions).
[22] H. Stewart, B. Wendroff, Two-phase flow: models and methods, Journal of Computational Physics 56 (1984) 363–409.
[23] D.S. Miller, G.B. Zimmerman, An algorithm for time evolving volume fractions in mixed zones in Lagrangian hydrodynamics calculations, Tech. Rep. UCRL-PRES-223908, Lawrence Livermore National Laboratory, 2006.
[24] A. Murrone, H. Guillard, A five equation reduced model for compressible two phase flow problems, Journal of Computational Physics 202 (2005) 664–698.
[25] P. Knupp, L. Margolin, M. Shashkov, Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian–Eulerian method, Journal of Computational Physics 176 (2002) 93–128.
[26] L. Margolin, M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grids, Journal of Computational Physics 184 (2003) 266–298.
[27] R. Loubere, M. Shashkov, A subcell remapping method on staggered polygonal grids for arbitrary Lagrangian–Eulerian methods, Journal of Computational Physics 209 (2005) 105–138.
[28] R. Garimella, M. Kucharik, M. Shashkov, An efficient linearity and bound preserving conservative interpolation (remapping) on polyhedral meshes, Computers and Fluids 36 (2007) 224–237.
[29] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, Journal of Computational Physics 39 (1981) 201–225.
[30] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, Journal of Computational Physics 121 (1998) 112–152.
[31] D.J. Benson, Volume of fluid interface reconstruction methods for multi-material problems, Applied Mechanics Reviews 55 (2002) 151–165.
[32] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, Journal of Computational Physics 100 (1992) 25–37.
[33] J. Glimm, J.W. Grove, X.L. Li, K. Shyue, Y. Zeng, Q. Zhang, Three-dimensional front tracking, SIAM Journal on Scientific Computing 19 (1998) 703–727.
[34] G. Tryggvasona, B. Bunnerb, A. Esmaeelic, D. Juricd, N. Al-Rawahic, W. Tauberc, J. Hanc, S. Nase, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, Journal of Computational Physics 169 (2001) 708–759.
[35] M. Sussman, E. Fatemi, P. Smereka, S. Osher, Improved level set method for incompressible two-phase flows, Computers & Fluids 27 (1998) 663–680.
[36] M. Sussman, A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, M.L. Welcome, An adaptive level set approach for incompressible two-phase flows, Journal of Computational Physics 148 (1999) 81–124.
[37] S. Osher, R.P. Fedkiw, Level set methods: an overview and some recent results, Journal of Computational Physics 169 (2001) 463–502.
[38] 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 (1998) 343–360.
[39] A. Stagg, R. Boss, J. Grove, N. Morgan, Interface modeling: a survey of methods with recommendations, Tech. Rep. LA-UR-05-8157, Los Alamos National Laboratory, 2005.
[40] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annual Review of Fluid Mechanics 31 (1999) 567–603.
[41] M. Rudman, Volume tracking methods for interfacial flow calculations, International Journal for Numerical Methods in Fluids 24 (1997) 671–691.
[42] J.A. Harte, W.E. Alley, D.S. Bailey, J.L. Eddleman, G.B. Zimmerman, Lasnex – a 2D physics code for modeling icf, Tech. Rep. UCRL-LR-105821-96-4, Lawrence Livermore National Laboratory, 1996.
[43] D. Bailey, S. Brown, G. Zimmerman, Interface reconstruction and sub-zone physics models, Tech. Rep. UCRL-CONF-214875, Lawrence Livermore National Laboratory, 2005.
[44] T.T. Team, *Truchas* physics and algorithms, Tech. Rep. LA-UR-06-1379, Los Alamos National Laboratory, 2006.
[45] D.L. Youngs, Numerical Methods for Fluid Dynamics, Academic Press, New York, 1982 (Ch. Time-dependent multi-material flow with large fluid distortion, pp. 273–285).
[46] D. Fressmann, P. Wriggers, Advection approaches for single- and multi-material arbitrary Lagrangian–Eulerian finite element procedures, Computational Mechanics 39 (2007) 153–190.
[47] S. Mosso, S. Clancy, A geometrically derived priority system for youngs' interface reconstruction, Tech. Rep. LA-CP-95-0081, Los Alamos National Laboratory, 1995.
[48] G.H. Miller, P.A. Colella, A conservative three-dimensional Eulerian method for coupled solid–fluid shock capturing, Journal of Computational Physics 183 (2002) 26–82.
[49] K.S. Bonnel, M.A. Ducheineau, D.R. Schikore, B. Hamann, K.I. Joy, Material interface reconstruction, IEEE Transactions on Visualization and Computer Graphics 9 (2003) 500–511.
[50] S.P. Schofield, R.V. Garimella, M.M. Francois, R. Loubere, Mutli-material interface reconstruction using particles and power diagrams, Tech. Rep. LA-UR-06-8740, Los Alamos National Laboratory, 2006.

[51] E.G. Puckett, J.S. Saltzman, A 3D adaptive mesh refinement algorithm for multimaterial gas dynamics, Physica D 60 (1992) 84–93.
[52] B.Y. Choi, M. Bussmann, A piecewise linear approach to volume tracking a triple point, International Journal for Numerical Methods in Fluids 53 (2006) 1005–1018.
[53] P. Liovic, M. Rudman, J.-L. Liow, D. Lakehal, D. Kothe, A 3D unsplit-advection volume tracking algorithm with planarity-preserving interface reconstruction, Computers & Fluids 35 (2006) 1011–1032.
[54] B. Swartz, The second-order sharpening of blurred smooth borders, Mathematics of Computation 52 (1989) 675–714.
[55] S.J. Mosso, B.K. Swartz, D.B. Kothe, R.C. Ferrel, A parallel, volume-tracking algorithm for unstructured meshes, in: P. Schiano (Ed.), Parallel Computational Fluid Dynamics '96, Italy, 1996.
[56] D.B. Kothe, M.W. Williams, K.L. Lam, D.R. Korzekwa, P.K. Tubesing, E.G. Puckett, A second-order accurate, linearity-preserving volume tracking algorithm for free surface flows on 3D unstructured meshes, in: Proceedings of the Third ASME/JSME Joint Fluid Engineering Conference, San Francisco, CA, USA, fEDSM99-7109, 1999.
[57] D.M. Gao, A three-dimensional hybrid finite element-volume tracking model for mould filling in casting processes, International Journal for Numerical Methods in Fluids 29 (1999) 877–895.
[58] H.T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, Tech. Rep. LA-UR-07-0656, Los Alamos National Laboratory, 2007. Also available as http://cnls.lanl.gov/~shashkov/.
[59] B. Mirtich, Fast and accurate computation of polyhedral mass properties, Journal of Graphics Tools 1 (1996) 31–50.
[60] E. Saltel, 3D meshes research database by eric saltel inria gamma group. http://www-c.inria.fr/gamma/gamma.php.
[61] C. Geuzaine, J.-F. Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. http://www.geuz.org/gmsh/.
[62] E. Vitali, D.J. Benson, An extended finite element formulation for contact in multi-material arbitrary Lagrangian–Eulerian calculations, International Journal for Numerical Methods in Engineering 67 (2006) 1420–1444.
[63] CUBIT: geometry and mesh generation toolkit. http://cubit.sandia.gov/.
[64] A. Danilov, Y. Vassilevski, aft3D meshing library, private communication.