Journal of Computational Physics 228 (2009) 731–745

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A second-order accurate material-order-independent interface reconstruction technique for multi-material flow simulations

Samuel P. Schofield [a,*], Rao V. Garimella [a], Marianne M. Francois [b], Raphaël Loubère [c]

[a] *Mathematical Modeling and Analysis (T-7), Los Alamos National Laboratory, Los Alamos, NM 87545, United States*
[b] *Computational Physics and Methods (CCS-2), Los Alamos National Laboratory, Los Alamos, NM 87545, United States*
[c] *Mathematical Institute of Toulouse, CNRS, University of Toulouse, France*

ARTICLE INFO

Article history:
Received 17 January 2008
Received in revised form 18 September 2008
Accepted 29 September 2008
Available online 11 October 2008

Keywords:
Volume-of-fluid
Interface reconstruction
Multi-material flow
Material-order independence
Linear reconstruction
Centroids
Power diagrams

ABSTRACT

A new, second-order accurate, volume conservative, material-order-independent interface reconstruction method for multi-material flow simulations is presented. First, materials are located in multi-material computational cells using a piecewise linear reconstruction of the volume fraction function. These material locator points are then used as generators to reconstruct the interface with a weighted Voronoi diagram that matches the volume fractions. The interfaces are then improved by minimizing an objective function that smoothes interface normals while enforcing convexity and volume constraints for the pure material subcells. Convergence tests are shown demonstrating second-order accuracy. Static and dynamic examples are shown illustrating the superior performance of the method over existing material-order-dependent methods.

1. Introduction

Multi-material and multi-phase flows occur in a variety of natural phenomena and industrial processes. To accurately model such flows, it is essential to effectively capture and manage material interfaces. Due to their ability to strictly conserve mass, volume-of-fluid (VOF) methods using interface reconstruction are widely used in such simulations [1–4]. Originally developed by Hirt and Nichols [5], VOF methods do not explicitly track interfaces but rather track the volume of each material. When required, the interface position is computed using the volume fraction data. In a flow simulation, the volume fractions are updated by determining the flux of each material into or out of a computational cell although in multi-material compressible simulations, volume fractions may also be modified by mixture models like pressure equilibration [1]. Contemporary schemes use the reconstructed interface to obtain a better approximation to the material fluxes. Poor interface reconstruction directly affects material fluxes and can result in material being transported to the wrong locations and unphysical fragmentation of material.

Early VOF methods used a straight line aligned with a coordinate axis to partition the cell according to the material volume fractions [6]. Youngs [7,8] extended the method to permit the material interface to have an arbitrary orientation within the cell. Such methods, that allow a generally oriented interface within the cell, are referred to as piecewise linear interface calculation (PLIC) methods [3]. In Youngs' method, the outward normal of the interface separating a material from the rest of

[*] Corresponding author. Tel.: +1 505 606 1484; fax: +1 505 665 5757.
E-mail addresses: sams@lanl.gov (S.P. Schofield), rao@lanl.gov (R.V. Garimella), mmfran@lanl.gov (M.M. Francois), loubere@mip.ups-tlse.fr (R. Loubère).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.09.023

the cell is taken to be the negative gradient of the "volume fraction function." The "volume fraction function" is assumed to be a smooth function whose cell-centered values are given by the cell-wise material volume fractions. The gradient may be computed by finite-difference formulas [8], Green–Gauss formula [9] or a least-squares technique [3]. The interface is then defined as a line with the calculated normal that cuts off the correct volume of material from the computational cell. We will refer to all methods that compute the interface from the gradient of the volume fraction function as gradient-based methods. In general, gradient-based methods for interface reconstruction are only first-order accurate although they may exhibit nearly second-order accurate behavior for rectangular grids. However, there are extensions that make the reconstruction second-order accurate by smoothing the interface normals [10–12]. Finally, there is a class of methods that attempts to reconstruct curved interfaces but these methods are not in widespread use [13].

VOF–PLIC techniques have been successfully used to accurately simulate two-phase (or two-material) flows and free-surface flows in two and three dimensions. However, their application to flows involving three or more materials that come closer than the mesh spacing and even form junctions has been mostly *ad hoc*. Examples of such phenomena are flows of immiscible fluids (e.g. oil–water–gas), inertial confinement fusion (ICF), hypervelocity impact and penetration, dynamic compaction of multi-material powders [2] and detonation shock dynamics with multiple inert and energetic materials [14]. In the oil–water–gas flow example, the three-immiscible fluids can statically or dynamically exhibit a thin film structure along the interface (Fig. 1(a)) or form a contact line (triple point in 2D) at the intersection between the three fluids (Fig. 1(b)). In inertial confinement fusion (ICF), a spherical plastic target filled with deuterium–tritium (solid and gas, see Fig. 2) is compressed by laser-initiated X-rays while located within an experimental chamber called a hohlraum that is usually made of gold or tantalum. The resulting compression and possible fragmentation of the multi-material target imply that thin layers and multi-material junctions are expected in the simulation and must be properly resolved.

The most common extensions of PLIC to cells with more than two materials (multi-material cells) is to process materials one-by-one leading to a reconstruction that is strongly dependent on the order in which the materials are processed. In a strict sense, any cell with more than one material is a multi-material cell. However, we choose to distinguish two-material cells from cells with more than two materials by calling the latter multi-material cells. This is because we have to reconstruct the interface for only one material in a two material cell whereas we have to reconstruct multiple interfaces in a cell with three or more materials. In the "onion-skin" approach, each material interface is assumed to separate two materials and consists of a single line segment with both endpoints on the computational cell boundary. This form of reconstruction works only for simple layer structures [7,15] and even there it may create overlapping layers. A more general and correct approach is the "nested dissection" method [16,17], where each material is separated from the others in a specified order. In the method, a pure polygon (or polyhedron) representing the first material is marked out from the cell, leaving a mixed polygon for the remaining materials. Then, a polygon representing the second material is marked out from the mixed polygon and the process continues until the last material is processed.

With the "correct" material ordering, the interface reconstructed by one of the above methods is close to the correct configuration (Fig. 3(a)). However an incorrect ordering results in substantial degradation of the interface as shown Fig. 3(b) and (c). Sometimes, there may not be an ordering which will give the correct configuration by nested dissection (Fig. 4) or the correct ordering in one part of the domain may be incorrect in a different part. Two-material cells next to multi-material cells may also be affected by the order in which materials are processed as discussed in [18].

Finally, the presence of multiple materials in simulations creates special considerations for second-order accurate methods like LVIRA. In the LVIRA method [10], the interface normal within a cell is updated by minimizing an objective function for each cell, $\mathcal{C}_i$

$$F^{LVIRA} = \sum_{j\in\mathcal{N}_i} (f_m(\mathcal{C}_j) - \tilde{f}_m(\mathcal{C}_j, \mathbf{n}, b))^2 \tag{1}$$

where $\mathcal{N}_i$ are the vertex connected neighbors of cell $\mathcal{C}_i, f_m(\mathcal{C}_j)$ is the prescribed volume fraction for material m in cell j, and $\tilde{f}_m(\mathcal{C}_j, \mathbf{n}, b)$ is the volume fraction cut off in cell j by continuing the line with normal $\mathbf{n}$ and line constant b from cell i through cell j. However, in multi-material cells the volume of a cell cut off by extending the interface may not be representative of the error. This is because the procedure fails to account for the presence of an additional material behind the extended line, giving a large error in the volume fraction even when the interface normal is correct as illustrated in Fig. 5.

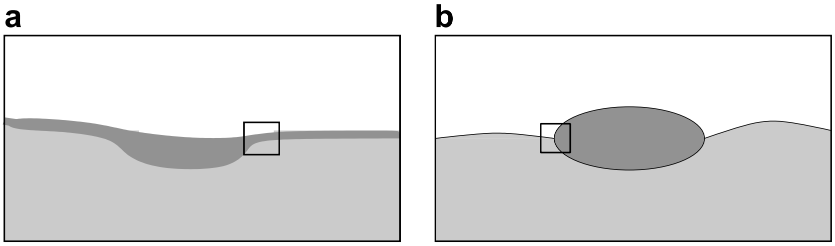

Fig. 1. Example of immiscible fluid interfaces (e.g. oil–water–air): (a) thin film and (b) triple point.

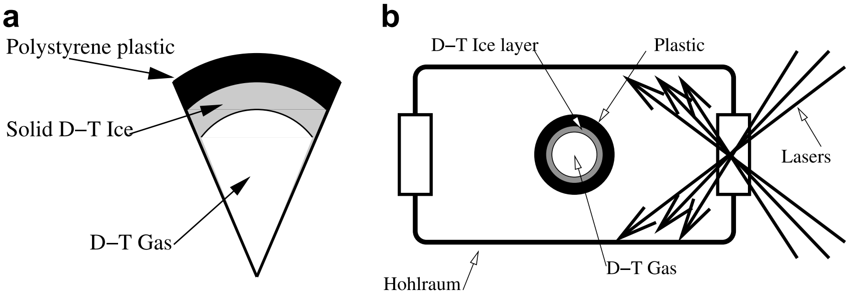

Fig. 2. (a) Sketch of an ICF target. (b) The target in the Holhraum (made of gold or tantalum), lasers enter the chamber generating X-ray emission that ultimately compress the target up to ignition of the deuterium–tritium gas.

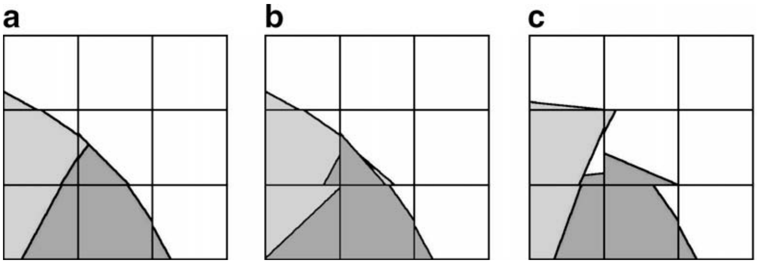

Fig. 3. (a) Reconstruction with the correct order, (b) and (c) reconstructions with an incorrect order.

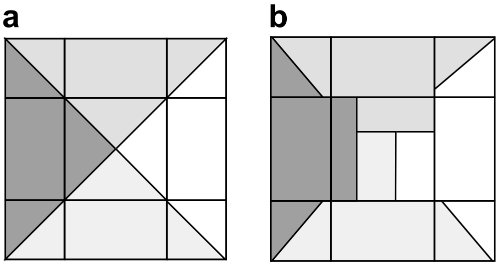

Fig. 4. (a) Four material junction and (b) order dependent reconstruction. In this example, the right configuration cannot be generated by any order of materials in a nested dissection process.

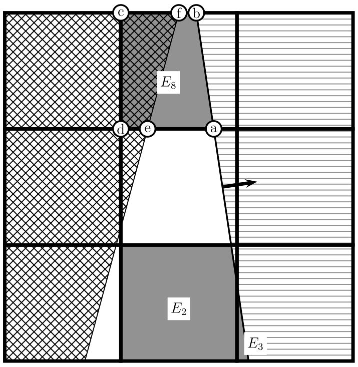

Fig. 5. In LVIRA, the interface normal for the white material in the center cell is adjusted by extending the interface from the center cell to the neighboring cells and minimizing the discrepancy between the volumes it cuts off in the neighboring cell, here labeled E_2, E_3, and E_8, and the prescribed volume fraction for that material in the neighboring cell. However, in computing the volume discrepancy the procedure may not be accounting for another material behind the interface as in $E8$ where the volume cutoff becomes the polygon `abcd` instead of `abfe`.

Most importantly, these incorrect reconstructions adversely impact the material advection in flow simulations. In most cases, an improper material ordering may result in materials being advected prematurely (or belatedly) into neighboring cells. This can further lead to small pieces of the material getting separated and drifting away from the bulk of the material, a phenomenon known as "flotsam and jetsam".

To address this problem, there has been some work on deriving the material order automatically. The geometrically derived material priority by Mosso and Clancy [19] first approximates the local center of mass of each material; then, assuming a layer structure, it selects an ordering based on the relative locations of the approximate centroids along a line. A similar approach was developed by Benson [20]. However, both methods are primarily designed to capture layers and may fail for multi-material junctions.

Choi and Bussman [15] have addressed the material ordering problem by developing a method for reconstructing three-material T-junctions in two-dimensions. In a three-material cell, the interface between material 1 and materials 2 and 3 are reconstructed as in a gradient-based method. Then the normal of the interface between materials 2 and 3 are determined by a minimization procedure similar to LVIRA. This method is not truly order-independent since the first material must be chosen *a priori* and it is restricted to three-material cells. Also, the feasibility of applying it to 3D is not clear.

Caboussat et al. [21] also addressed the reconstruction of a triple point using a minimization procedure given by the neighboring normals. However, it is also restricted to three materials.

Bonnell et al. [22] have also described an interface reconstruction method that draws upon ideas of isosurfacing. This method can handle multiple materials in a cell automatically but the method is not guaranteed to match volume fractions exactly.

Previously, we presented a method that reconstructed interfaces in a material-order independent manner using a particle attraction–repulsion model or a regional quadrature formula to locate materials in cells and subdividing the cell using a weighted Voronoi diagram or power diagram [18]. This method is capable of generating a material-order independent partitioning of multi-material cells but is not second-order accurate.

In this article, we present a new second-order accurate VOF–PLIC method that reconstructs a multi-material interface with no dependence on the order in which materials are specified. The method matches volume fractions exactly as required by VOF interface reconstruction methods. The method works for arbitrary number of materials in general polygonal meshes. The reconstruction automatically gives either the appropriate layer structure or configurations with multiple material junctions. Finally, the method ensures that the reconstructed interface partitions the cell into convex material regions, an important consideration for a majority of material advection and remapping schemes. Finally, it reproduces straight line interfaces exactly and reconstructs interfaces with second-order accuracy.

Unlike the particle model method or quadrature formula method of [18], the new technique infers relative locations of materials based on a piecewise linear approximation to the characteristic function of each material. Using the approximate material locations, the cell is partitioned into material regions by a weighted Voronoi diagram while matching the volume fractions. Finally, the interface segments in the multi-material cells are smoothed with respect to their neighbors so that the method reproduces smooth interfaces with second-order accuracy even around multi-material junctions.

The following sections describe the three main steps of the new method, i.e. material location, power diagram reconstruction and smoothing. These are followed by results demonstrating that the method performs better than material order-dependent methods in static, geometric reconstructions and in dynamic advection tests. Convergence tests are presented to demonstrate that the method is second-order accurate.

2. Material location by piecewise linear reconstruction of the volume fraction function

Given the volume fractions of materials in cells in a mesh, our task is to determine the relative locations of materials in a multi-material cell. To do this, we must ideally recover the characteristic function for each material in the domain. While it is possible to reconstruct the characteristic function in 1D [23], no method (other than interface reconstruction itself) exists to do this in higher dimensions. Therefore, we make a simplifying assumption that a smooth function, called the volume fraction function, exists for each material and that its pointwise cell-centered values are given by cell-wise volume fraction data. This smooth function represents the distribution of material in the mesh cells and in that sense, it can be considered analogous to a density distribution function for the material. However, we should note that the volume fraction function is not a clearly defined mesh-independent continuous function like the density function. Swartz [24] describes it as the function that quantifies the relative amount of a material present in a small window that moves around in a domain with a sharp interface. Defined this way, it is clear that the volume fraction function steepens as the size of the window (or in other words, the mesh size) gets smaller and the gradient of the function blows up as the window size goes to zero. Nevertheless, for a given mesh, we will treat the volume fraction function like a smooth, density distribution function.

We then compute a piecewise linear approximation for this smooth volume fraction function using standard methods used in higher-order finite-volume methods [9]. Finally, continuing the analogy with the density function, we compute the center of mass of the materials in cells from the linear reconstruction as described below.

Consider a mesh on which we have cell-centered values f_i of a function $f(\mathbf{x})$. In each cell $\mathcal{C}_i$, we reconstruct a linear approximation, $\tilde{f}_i(\mathbf{x})$, of the function such that

$$\tilde{f}_i(\mathbf{x}) = f_i + \nabla f \cdot (\mathbf{x} - \mathbf{x}_c(\mathcal{C}_i)), \tag{2}$$

where $\mathbf{x}_c(\mathcal{C}_i)$ is the centroid of the cell. ∇f is the gradient of the function that we wish to approximate and it is considered to be constant within the cell. The gradient may be computed either by a Green–Gauss [9] or a least-squares technique [3]. On structured and unstructured grids, we use all vertex and edge connected neighbors in the gradient computation. In Fig. 6, the path used for the Green–Gauss technique is shown with the dotted line. For a least-squares technique, the same mesh cells are used in the computation with each entry weighted by the inverse of the squared distance between the centroid of the cell being reconstructed and the centroid of the neighboring cell as described in [3]. The computed gradient is limited using Barth–Jesperson-type limiter [25] to preserve local bounds on the volume fraction function. The limiter is calculated using all vertex connected neighbors. The limited gradient is indicated by $\boldsymbol{\delta} = \phi\nabla f$ with $\phi \in [0,1]$. Then, the approximate center of mass of the function $f(\mathbf{x})$ over the domain Ω_i as approximated by the function $\tilde{f}_i(\mathbf{x})$ is given by

$$\bar{\mathbf{x}} = \frac{\int_{\Omega_i} \mathbf{x}\tilde{f}_i(\mathbf{x})d\Omega}{\int_{\Omega_i} \tilde{f}_i(\mathbf{x})d\Omega} = \frac{1}{\|\Omega_i\| f_i}\int_{\Omega_i} \mathbf{x}(f_i + \boldsymbol{\delta}\cdot(\mathbf{x} - \mathbf{x}_c(\mathcal{C}_i)))d\Omega \tag{3}$$

where $\|\Omega_i\|$ is the area of the domain Ω_i.

The obvious choice of domain Ω_i for integrating this equation is the cell, $\mathcal{C}_i$ and this works well for structured meshes. The calculation of Eq. (3) for a polygon may be done with the application of Stokes' theorem in the plane, for details see [26]. However, for unstructured meshes, we have found that integrating over the cell domain induces a strong bias in the orientation of the reconstructed interface based on the cell geometry. In order to eliminate this effect, we integrate instead over the smallest square, $S(\mathcal{C}_i) \supseteq \mathcal{C}_i$, whose center coincides with the centroid of the cell, $\mathbf{x}_c(\mathcal{C}_i)$ and encloses the computational cell. An example of this domain is shown with the solid line in Fig. 6.

For two materials, this choice of integration domain is equivalent to a gradient-based method when using a power diagram interface reconstruction. In a power diagram based reconstruction of a two material cell, the interface normal depends only on the direction of the vector pointing from one material locator to the other. For two materials, m and n, with material locators $\mathbf{x}_m$ and $\mathbf{x}_n$ and volume fractions f_m and $1 - f_m$, respectively, the normal to the interface between them given by the power diagram reconstruction will be

$$\mathbf{x}_m - \mathbf{x}_n = \frac{1}{\|S(\mathcal{C}_i)\|}\left(\frac{1}{\bar{f}_m} + \frac{1}{1-\bar{f}_m}\right)\int_{y_0}^{y_1}\int_{x_0}^{x_1} \mathbf{x}(\boldsymbol{\delta}\cdot(\mathbf{x} - \mathbf{x_c}))dx\,dy = \frac{\Delta^2}{12}\left(\frac{1}{\bar{f}_m(1-\bar{f}_m)}\right)\begin{pmatrix}\delta_x\\ \delta_y\end{pmatrix}$$

where $S(\Omega_i) = [x_0, x_1] \times [y_0, y_1]$ and $\Delta = x_1 - x_0 = y_1 - y_0 = \sqrt{\|S(\mathcal{C}_i)\|}$. That is, the normal is a positive constant times the gradient. Hence, the interface normal will be the computed gradient. In addition, this choice of integration domain makes the calculation of Eq. (3) trivial and provides a better initial reconstruction for a starting point to the interface smoothing procedures.

For meshes and simulations with special symmetries such as uniform-in-angle grids and problems with cylindrical symmetry [27], the choice of the square integration domain may break symmetry. In such cases, a different integration domain, such as the mesh cell itself, may be preferable.

3. Power diagram based interface reconstruction

Once the materials are located in a cell, the interfaces within the cell, separating the materials, are constructed using a power diagram. A power diagram or Laguerre diagram [28,29] is a generalized Voronoi diagram of a set of points, S, each with an associated weight, ω_i. In this context, the generators are the material locators determined by the method detailed in Section 2.

The power of a point $\mathbf{x} \in \mathbb{R}^n$ with respect to a point mass, $s_i \in S$ with $s_i = (\mathbf{x_i}, \omega_i)$ is defined as

$$\text{pow}(\mathbf{x}, s_i) = d^2(\mathbf{x}, \mathbf{x_i}) - \omega_i^2 \tag{4}$$

where $d^2(\mathbf{x}, \mathbf{x_i}) = \sum_{j=1}^{n}(x^j - x_i^j)^2$ is the usual Euclidean distance in $\mathbb{R}^n$.

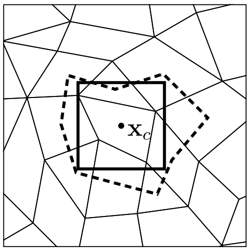

Fig. 6. The dotted line shows the path used for the Green–Gauss computation of the gradient in the center cell. The dark solid line shows the square domain, $S(\Omega_i)$, used for the material location calculation. It is the smallest square that covers the center cell and has the same centroid, $\mathbf{x}_c$, as the center cell.

Each cell of the power diagram is the set of points

$$\mathrm{cell}(s_i) = \{\mathbf{x} \in \mathbb{R}^n | \mathrm{pow}(\mathbf{x}, s_i) < \mathrm{pow}(\mathbf{x}, s_j) \forall s_j \in S, s_j \neq s_i\}. \tag{5}$$

As with Voronoi diagrams, each power diagram cell is convex with a piecewise linear boundary. If all the weights are equal, the power diagram degenerates to a Voronoi diagram. The weight associated with a point generator can be interpreted as the square of the radius of a circle centered at that point. An example power diagram is shown in Fig. 7.

A power diagram can be constructed by a randomized incremental algorithm [30] or by conversion from a Voronoi diagram [31] which may be created using a number of efficient algorithms such as Fortune's sweepline algorithm [32]. However, we use a simple half-space intersection algorithm because it is simple and robust for a small number of materials in each cell. The half-space intersection algorithm computes the power diagram cell for each generator by clipping the polygon by each bisector between that generator and all other generators, starting with the original computational cell.

3.1. Matching volume fractions

The volume fractions of the materials in a cell are matched by iteratively adjusting the weights of the point generators in a power diagram, thereby adjusting the area of the power diagram cells once clipped to the mesh cell. This requires the solution of a set of non-linear equations

$$A_m(\omega_1^2, \ldots, \omega_{N_m}^2) = \|\mathcal{C}_i\| f_m, \quad m = 1, \ldots, N_m \tag{6}$$

where $A_m(\omega_1^2, \ldots, \omega_{N_m}^2)$ is the area of the power diagram corresponding to material m after it has been clipped by the bounding mesh cell polygon with area $\|\mathcal{C}_i\|$. f_m is the volume fraction for material m in cell $\mathcal{C}_i$. The constraint:

$$\sum_m^{N_m} A_m(\omega_1^2, \ldots, \omega_n^2) = \|\mathcal{C}_i\| \tag{7}$$

reduces the number of equations to $N_m - 1$. Since for any power diagram

$$A_m(\omega_1^2 + c^2, \ldots, \omega_n^2 + c^2) = A_m(\omega_1^2, \ldots, \omega_n^2) \tag{8}$$

for any real number c, a normalized set of weights must be chosen. Specifically, this is done by forcing one of the weights to be a specified value and varying the others.

A Newton procedure with a finite-difference Jacobian is used to solve Eqs. (6) and (7). At each iteration of the Newton procedure, a power diagram, clipped to the mesh cell is computed to determine the A_m and approximate the Jacobian. Some caution is required, since the area of each cell is bounded above and below, that is

$$0 \leqslant A_m(\omega_1^2, \ldots, \omega_n^2) \leqslant \|\mathcal{C}_i\| m = 1, \ldots, N_m \tag{9}$$

For extreme values of the weights, some of the power diagram cells will be outside of the mesh cell and as such have zero area once clipped to the mesh cell, $\mathcal{C}_i$, under consideration. In such cases, one or more of $A_m(\omega_1^2, \ldots, \omega_n^2)$ will be constant with value 0 or $\|\mathcal{C}_i\|$ and as a function of the weights $(\omega_1^2, \ldots, \omega_n^2)$ it will have a zero gradient making a straightforward Newton procedure fail. As a result, the Newton procedure needs to adjust for overshoots to make sure it does not end up in this region. This is simply done by reducing the size of the Newton step at each iteration if it exceeds those bounds. We found the procedure to be robust and efficient, typically matching the required volumes to within 10^{-12} in 3–6 iterations.

For the initial guess, we use equal weights for all the point generators if all of the generators lie within the mesh cell being reconstructed. If any of the generators are outside the cell the initial weights are assigned such that the power bisectors be-

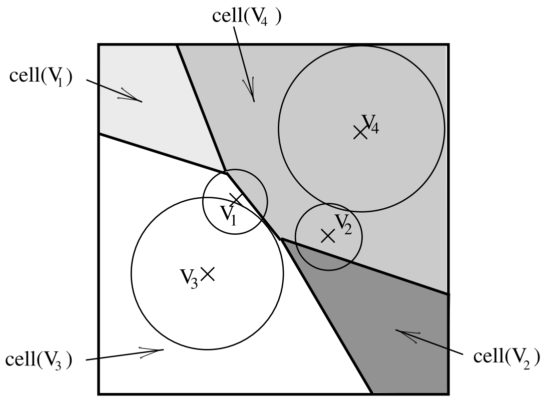

Fig. 7. Power diagram with four generators (V_1, V_2, V_3, and V_4) and their weight circles. The generators V_1 and V_2 do not lie within their corresponding cells.

tween all the generators go through the centroid of the cell. This ensures that the initial power diagram will have non-zero areas for each power diagram cell once clipped to the mesh cell.

4. Interface smoothing

Once an initial reconstruction of the interface is available, we apply an order-independent optimization-based smoothing procedure to improve the approximation of the interface.

Consider a 2D cell with N_m materials, N_s interface segments and N_p interface points. The smoothing procedure repositions the cell's N_p interface points so that it minimizes the discrepancy between the normal of each of its interface segments and normals of reference interface segments in neighboring cells (separating the same materials). The constraints imposed on this process are that the volume fractions of the materials in the cells must be matched exactly and that all the pure material subcells remain convex. Naturally, interface points on the boundary of the cell must remain on the boundary and interior points must remain strictly inside the cell.

The local objective function for smoothing in a particular cell, $\mathcal{C}_i$ is written as

$$F_i(\mathbf{s}) = \sum_{j=1}^{N_s} \sum_{k=1}^{(N_r)_j} \|\hat{\mathbf{n}}_j(\mathbf{s}) - \hat{\mathbf{n}}_k^r\|^2 \tag{10}$$

In the above equation, $\hat{\mathbf{n}}_j$ is the normal of the jth interface segment, $\hat{\mathbf{n}}_k^r$ is the normal of the kth reference segment, and finally, $(N_r)_j = 0, 1$ or 2 is the number of reference segments for the jth interface segment. Also, $\mathbf{s}$ is the vector of optimization variables which includes the x, y coordinates of interior interface points and 1D parameters of points on cell edges.

The reference segments for an interface segment are chosen according to the smoothing procedure laid out by Swartz [24] for smoothing two material interfaces. Given an interface segment in cell $\mathcal{C}_i$, we first find its endpoints that lie on the boundary of the cell. If the endpoint lies on a cell edge, we find the neighboring cell on the other side of the edge. In this neighboring cell, we attempt to find another interface segment that separates the same two materials as the interface segment in $\mathcal{C}_i$ that is under consideration. The reference normal for the interface segment is then the normal of the line segment connecting the midpoints of the two interface segments (see Fig. 8). Using this normal leads to a second-order accurate method for smooth interfaces in Swartz's work. If a neighboring cell does not exist (because the edge is on the domain boundary) or a suitable interface segment does not exist in the neighboring cell, we attempt to find a suitable reference segment in a cell connected to the vertices of the cell edge that the interface endpoint lies on. If the endpoint of an interface segment lies on a vertex, the reference interface segment is chosen from one of the vertex connected neighbors. Thus, the algorithm will pick two, one or zero reference segments for each interface segment. While it is possible that there may be more than two neighboring mixed cells with candidate reference segments, such a situation usually represents noisy volume fraction data or a feature of the flow that is not well resolved; in such situations, no perfect solution exists and therefore, any choice will probably lead to unsatisfactory results.

The formulation of the objective function is illustrated with the help of the examples shown in Fig. 8. The first example (Fig. 8(a)) shows a two-material cell of interest (shaded). Since there is only one interface segment with normal $\hat{\mathbf{n}}_1$ and its endpoints on the cell boundaries, this interface segment has two reference normals. Therefore, the objective function is

$$F_i(\mathbf{s}) = \|\hat{\mathbf{n}}_1(\mathbf{s}) - \hat{\mathbf{n}}_1^r\|^2 + \|\hat{\mathbf{n}}_1(\mathbf{s}) - \hat{\mathbf{n}}_2^r\|^2 \tag{11}$$

and the optimization variables are the 1D parameters of points $\mathbf{x}_1$ and $\mathbf{x}_2$. In the second example (Fig. 8(b)), there are three interface segments with normals $\hat{\mathbf{n}}_1, \hat{\mathbf{n}}_2$ and $\hat{\mathbf{n}}_3$ and endpoints $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ and $\mathbf{x}_4$. Therefore, the objective function is

$$F_i(\mathbf{s}) = \|\hat{\mathbf{n}}_1(\mathbf{s}) - \hat{\mathbf{n}}_1^r\|^2 + \|\hat{\mathbf{n}}_2(\mathbf{s}) - \hat{\mathbf{n}}_2^r\|^2 + \|\hat{\mathbf{n}}_3(\mathbf{s}) - \hat{\mathbf{n}}_3^r\|^2 \tag{12}$$

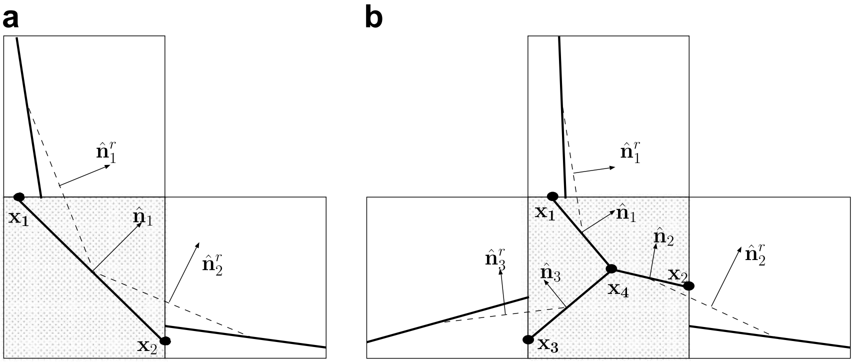

Fig. 8. Definition of reference normals for smoothing in (a) two-material cell and (b) three-material cell. The cell of interest is shaded.

the optimization variables are the 1D parameters of boundary points $\mathbf{x}_1, \mathbf{x}_2$ and $\mathbf{x}_3$, and the x, y coordinates of the interior point, $\mathbf{x}_4$.

The volume conservation constraint for the optimization can be expressed in the form of an equality constraint as

$$G_i(\mathbf{s}) = \frac{1}{\|\mathcal{C}_i\|^2} \sum_{m=1}^{N_m} (V_m(\mathbf{x}) - V_m^r)^2 = 0 \tag{13}$$

where $V_m(\mathbf{x})$ is the computed volume and V_m^r is the reference or specified volume of the mth material.

The convexity constraint on the subpolygons is expressed as a constraint on all interior angles of the subpolygons in the cell. The constraint function for convexity, $H_i(\mathbf{s})$ is a step function which is zero if all the interior angles of all material polygons in the cell are greater than 0 but less than or equal to π, and a very large penalty number (10^6) otherwise.

The optimization is then set up to minimize the composite function:

$$F_i^*(\mathbf{s}) = F_i(\mathbf{s}) + \lambda G_i(\mathbf{s}) + H_i(\mathbf{s}) \tag{14}$$

where λ is a penalty parameter. When the penalty parameter is sufficiently large, the constraints are enforced exactly [33].

The procedure minimizes $F_i^*(\mathbf{s})$ in each cell using a multi-dimensional non-linear conjugate gradient minimization procedure [33] with increasing values of the penalty parameters. Since the original reconstruction will satisfy the volume and convexity constraints, the penalty parameter λ is initially set to one, then incrementally increased by multiplying by 10 and smoothing using that value in the objective function until the penalty parameter reaches a desired value (typically 10^5).

For two materials cells the minimum of the objective functional in Eq. (14) may be constructed explicitly. It is the reconstruction corresponding to a single line with a normal taken by averaging the reference normals and matching the volume fraction. The multi-dimensional minimization procedure does not need to be invoked.

Multiple iterations ($N_{global} = 3$ to 5) over all mixed cells are performed so as to minimize the global objective function that can be formed by summing the F^*s over all mixed cells. The optimization converges to a correct solution quickly in all but a few very difficult cases for which convergence is impractically slow. The overall minimization algorithm is given in pseudocode in Fig. 9.

In Fig. 10, the initial power diagram reconstruction is shown along with the material locators for a three material, curved interface. After smoothing, the interface is much improved.

5. Numerical experiments

In this section, we present some static interface reconstruction examples and demonstrate second-order convergence. To demonstrate that our method leads to significant improvements in dynamic simulations, we present results of advection of multi-material configurations on structured and unstructured meshes.

5.1. Interface reconstruction

To demonstrate the performance of the method, two multi-material interface reconstructions are shown in Fig. 11. Both examples are on a domain of $[0,1] \times [0,1]$.

The first row in Fig. 11 shows the reconstruction of a triple point. In the center cell, the volume fractions are (0.5, 0.3, 0.2). In the top center and bottom center cells the volume fractions are (0.5, 0.0, 0.5) and (0.5, 0.5, 0.0), respectively. The right center cell has volume fractions of (0.0, 0.6, 0.4).

```
Material order independent smoothing
$\mathcal{I}^1$ := power diagram reconstruction
For $n = 1$ to $N_{global}$
    For each mixed cell $\mathcal{C}_i$
        construct $F^*(\mathbf{s}^0)$ using $\mathcal{I}^n$
        For $k = 0$ to 5
            $\lambda := 10^k$
            $\mathbf{s}^{k+1}(\mathcal{C}_i) := minimizer F^*(\mathbf{s}^k; \lambda)$
        End
        $\mathcal{I}^{n+1}(\mathcal{C}_i) =$ interfaces$(F^*, \mathbf{s}^K)$
    End
End
```

Fig. 9. Material order independent interface smoothing algorithm. Here $\mathcal{I}^n$ is the interface reconstruction at global iteration n. $\mathbf{s}^k$ is the vector of optimization variables at each stage of the inner loop which increases the penalty parameter λ. *minimizerF*$^*(\mathbf{s}^k; \lambda)$ returns the variables at the local minimum starting from initial guess $\mathbf{s}^k$. The routine **interfaces**$(F^*, \mathbf{s})$ gives the polygonal decomposition of the cell corresponding to the values of the optimization variables, $\mathbf{s}$.

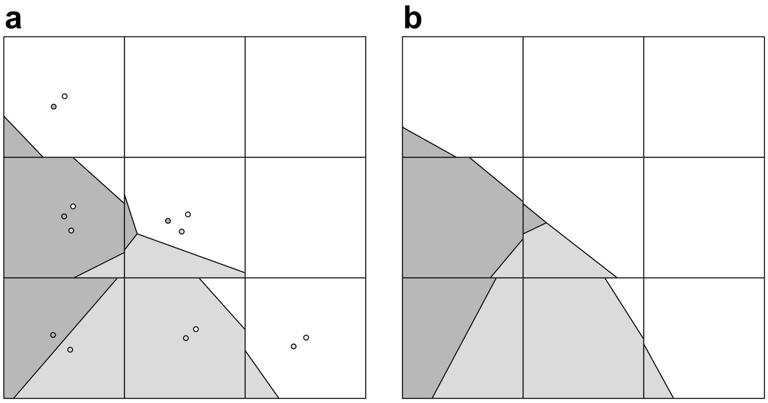

Fig. 10. (a) Interface reconstruction before smoothing showing material locators and (b) interface reconstruction after smoothing.

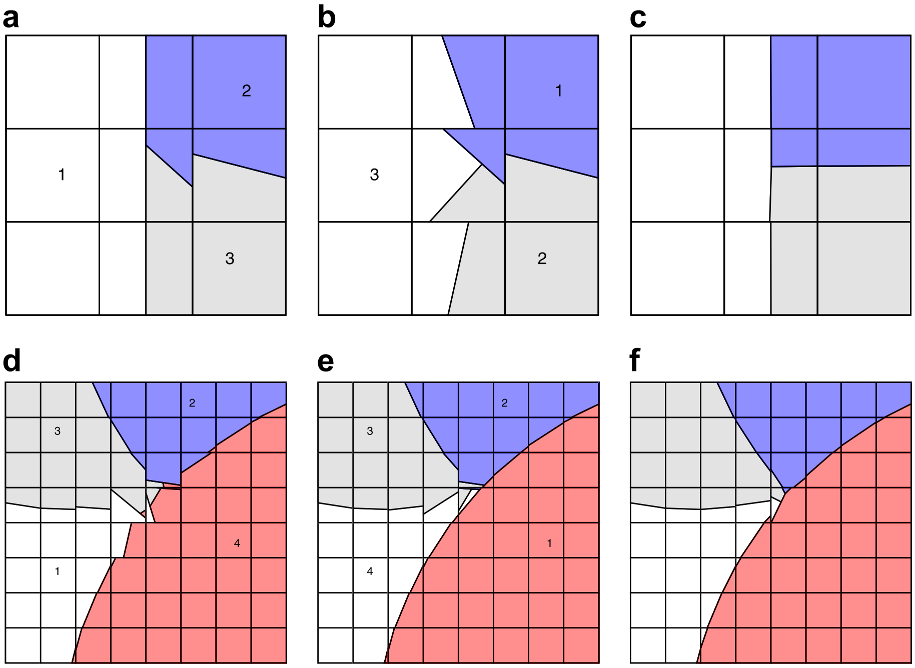

Fig. 11. Static interface reconstruction tests. (a) and (b) Triple point configuration using a nested dissection/gradient-based method with the material ordering indicated in the figure. (c) Triple point configuration using our second order method. (d) and (e) Three circle intersection test using nested dissection method with two different material orderings. (f) The interface reconstruction using our second order method. The convexity restrictions and initial topology of the power diagram prevent the optimization from completely smoothing the interface in the four material cells.

The second row of Fig. 11 shows the reconstruction of a three circle configuration containing two, three and four material cells. The volume fractions are defined by three circles given by (A) radius 0.5385, center (0.3, 1.0) (B) radius 0.7071, center (0.99, 0.97) and (C) radius 0.7071, center (0.99, 0.01) with circle (C) overlaying (B) which overlays (A).

The method correctly reconstructs the triple point in Fig. 11(c), although the white region in the center cell is only marginally convex due to the node at the center of the T-junction. Note in Fig. 11(d) and (e), the effect of multiple materials on a gradient-based method even in the two material cells near the four material center cell.

In Fig. 11(f), the method significantly improves the interface reconstruction in the four material and surrounding cells. However, due to the convexity constraints and the initial topology of the power diagram reconstruction the method is unable to perfectly match the interfaces in the four material cell.

5.2. Convergence

A number of tests have been conducted to demonstrate that the method converges and is second-order accurate. The error in each cell is measured as the symmetric area difference between the approximate and "true" reconstructions. Given two shapes Ω_1 and Ω_2, the symmetric area difference is measured as

$$e = (\|\Omega_1\| - \|\Omega_1 \cap \Omega_2\|) + (\|\Omega_2\| - \|\Omega_1 \cap \Omega_2\|) \tag{15}$$

The overall error, E, in the reconstruction is obtained by summing e over all materials and all cells. The symmetric area difference for a single cell is illustrated in Fig. 12.

The volume fraction initialization uses a sampling method based on adaptive mesh refinement. To compute the volume fractions of materials in a cell, the vertices of the cell are assigned material IDs based on an in-out test. If the vertices of the cell are of the same material, the cell is assigned that material, If not, the cell has multiple materials and is then subdivided into four equal subcells (assuming the cell is a regular quadrilateral). The new points generated by refinement are assigned material IDs based on further in-out tests. The refinement continues in this way until all subcells are pure or the size of a subcell is below the required volume fraction tolerance. The areas of subcells are then added up to compute volume fractions up to the desired tolerance.

Three multi-material examples where chosen for the convergence tests. The problem domain was $[0,1] \times [0,1]$ for all three tests.

- The first test (*circle*) shown in Fig. 13(a) consists of a circle of radius 0.25 with center (0.5, 0.5).
- The second test (*two-arcs*) shown in Fig. 13(b) is a three-material problem with two intersecting circular arcs. The bottom circle has a radius of 0.7 and center (0.5, 0.0). The circle on the right overlays the bottom circle and has radius of 0.5 and center (1.0, 0.5).
- The third test (*curved-filament*) is a curved filament with a varying thickness as shown in (Fig. 13(c)). It is composed of two circles, one with a radius of 0.7 and center (0.5, 0.0) which is partially covered by a circle of radius 0.64 and center (0.5, 0.05).

The coarsest mesh was a regular mesh of 32 × 32 and the finest mesh was 256 × 256, which had a characteristic mesh spacing of $h = \frac{1}{256}$.

The errors for these examples along with errors of an exactly second-order accurate scheme are plotted in Fig. 14. From this plot, it can be seen that the scheme reconstructs interfaces with second-order accuracy.

5.3. Advection

Interface reconstruction methods are generally applied in dynamic simulations where the materials, as described by the volume fractions, move over time. In simple, multiple passive scalar transport each material satisfies an advection equation,

$$\frac{d}{dt} f^m(\mathcal{C}_i) = -\frac{1}{\|\mathcal{C}_i\|} \int_{\partial \mathcal{C}_i} \chi_m(\mathbf{x}) \mathbf{u}(\mathbf{x}) \cdot \mathbf{n} d\mathbf{x} \tag{16}$$

where $\chi_m(\mathbf{x})$ is the characteristic function for material m and $\mathbf{u}$ is the prescribed transport velocity and $\mathbf{n}(\mathbf{x})$ is the unit outward normal on the cell boundary. Furthermore, we assume the velocity field is divergence free and the materials are immiscible but with no surface tension.

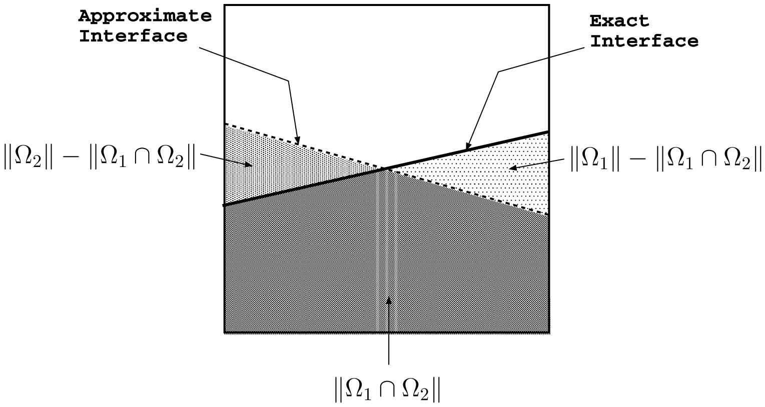
Approximate Interface
Exact Interface
$\|\Omega_2\| - \|\Omega_1 \cap \Omega_2\|$
$\|\Omega_1\| - \|\Omega_1 \cap \Omega_2\|$
$\|\Omega_1 \cap \Omega_2\|$

Fig. 12. Illustration of symmetric area difference with exact and approximate straight line interfaces. The darkly shaded part shows the area of overlap between the areas behind the two reconstructions and the remaining shaded parts show the area of non-overlap.

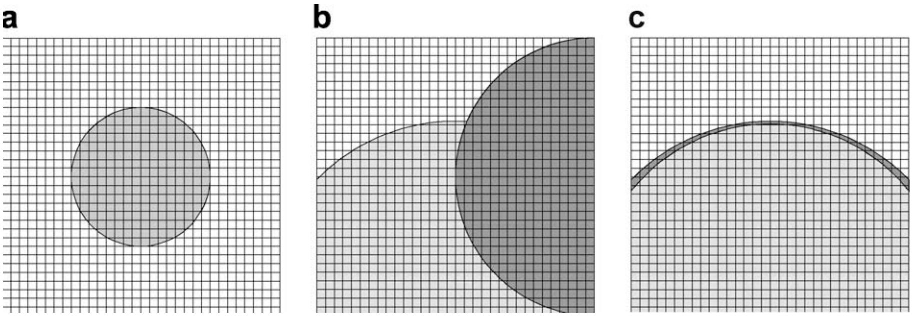

Fig. 13. Test cases used to study convergence – (a) two-material circular interface (*circle*) (b) three-material junction with circular arcs (*two-arcs*) and (c) varying thickness curved filament (*curved-filament*).

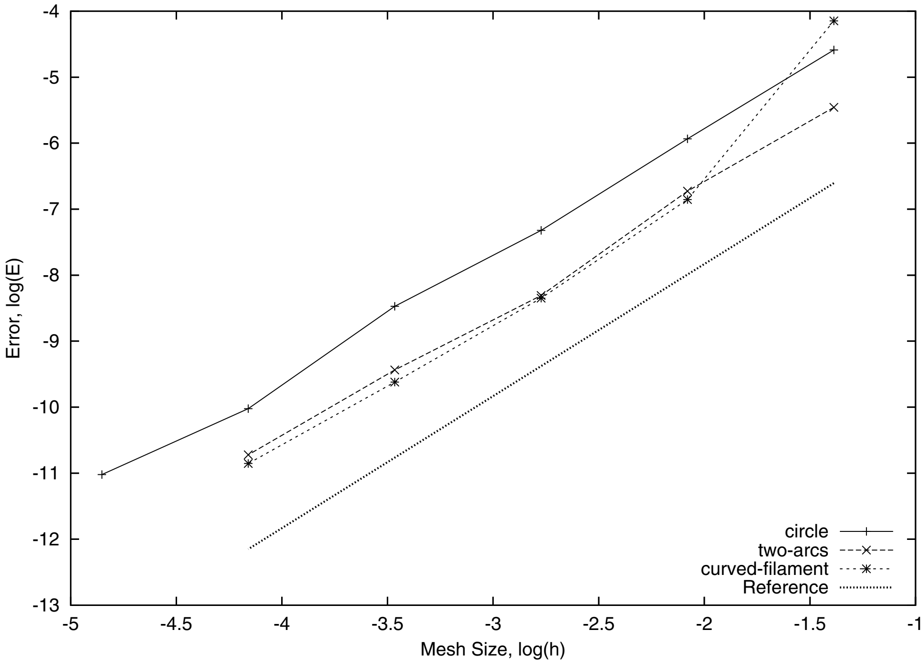

Fig. 14. Convergence plots for the three examples showing second-order accuracy. Here h is the mesh spacing and E the overall error.

A number of material advection schemes exist to simulate Eq. (16). For unstructured grids, an attractive advection scheme is the Lagrangian step plus Eulerian remap approach [34]. In this scheme, the advection step consists of a Lagrangian phase, where the mesh cell is moved in a Lagrangian fashion, followed by a conservative remap, where the cell contents from the Lagrangian step are mapped back to the Eulerian mesh. This approach has been suggested for material advection by a number of authors including [11,35–37].

There are two basic formulations of this method: a forward trajectory remap or a backward trajectory remap. In the forward trajectory remap, the position of the cell is calculated at time t^{n+1} and the interface reconstruction is performed using the updated cell positions. The material interfaces are then intersected with the original Cartesian mesh to determine the updated material quantities.

In the backward trajectory remap method, used in this work, the Eulerian mesh is taken as the cell positions at time t^{n+1}. The position of the cell is then integrated backwards in time to determine its location at the previous time step, t^n, which we refer to as the "preimage" of the cell. The interface reconstruction is performed using the volume fractions at time t^n on the original mesh. The preimage of the cell is then intersected with the interfaces on the Eulerian mesh. This process is illustrated in Fig. 15. The primary advantage of the backward trajectory remap, is that the interface reconstruction is always performed on the same, usually more regular, Eulerian mesh unlike the forward method where the interface reconstruction is performed on the Lagrangian mesh, which may consist of deformed cells even if the original Eulerian mesh was structured.

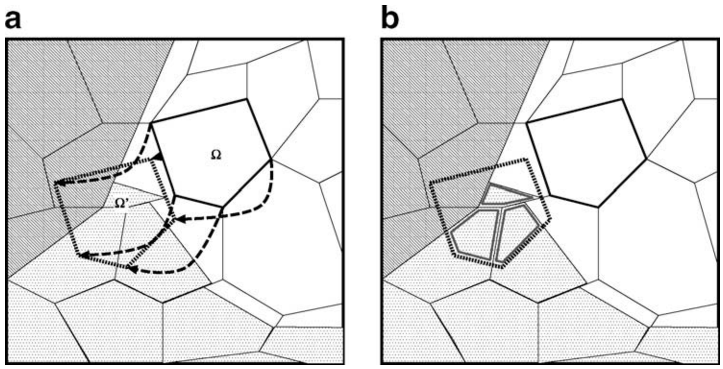

Fig. 15. Backward Lagrangian advection scheme. (a) The “preimage”, $\Omega\prime$ of the center computational cell of the fixed Eulerian mesh, obtained by integrating the nodes backward in time. (b) To determine the cell contents at the new time, the “preimage” is intersected with the interface reconstruction for each material at the previous time.

In addition, the computation of the gradient of the volume fraction function depends on cell geometry which is fixed for the backward scheme, but may change each timestep in the Lagrangian phase using the forward method.

In the first step of the backward trajectory remap method, the position of the cell vertices, $\mathbf{x}_i$, at the previous time step are determined by integrating the ODE backwards in time

$$\frac{d\mathbf{x}_\mathbf{i}^\mathbf{n}}{dt} = \mathbf{u}(\mathbf{x},t)t \in [t^{n+1},t^n] \tag{17}$$

with initial condition

$$\mathbf{x}_i^{n+1}(t^{n+1}) = \mathbf{x}_i \tag{18}$$

using the computed velocities from times t^n and t^{n+1}. We use a fourth order Runge–Kutta method, although many other time integrators are suitable.

Once the location of the vertices at the previous time step are calculated, the preimage of the cell is taken to be the polygon consisting of those vertices connected with straight lines. There are some errors introduced in computing the preimage in this manner. For a solenoidal velocity field $(\nabla \cdot \mathbf{u} = 0)$, the area of the preimage will be the same as the area of the original cell. For a general velocity field, the preimage of a line will typically be a curve. Failing to account for the curvature of the edges of the preimage introduces a defect in the cell area estimated to be $\mathcal{O}(h^3 \Delta t)$ in addition to area defects associated with backward time integration of the vertices.

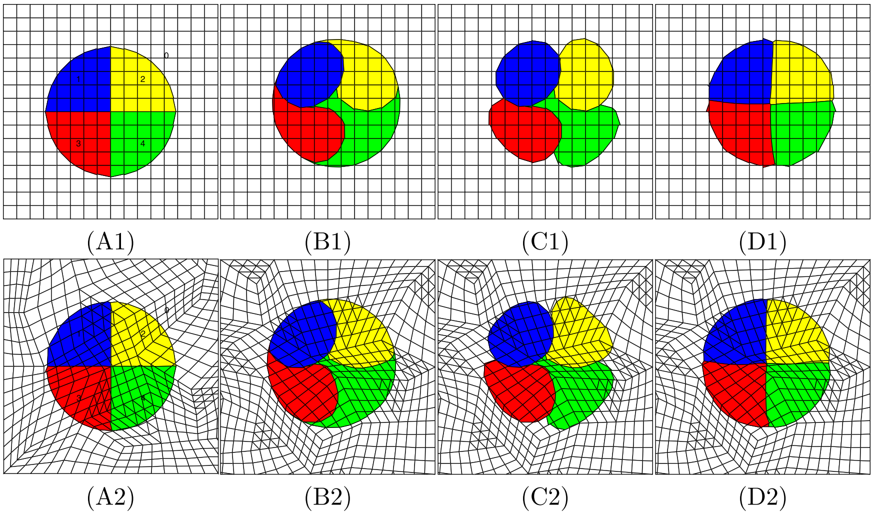

Fig. 16. Final configuration of the four material circle shown in (A1) with material numbers after diagonal translation with a velocity of (1, 1) at time $t = 0.5$ using the interface reconstruction methods: (B1) Youngs’ with material order (0, 1, 2, 3, 4) (C1) Youngs’ with material order (1, 2, 3, 4, 0) (D1) our second order method. (A2)–(D2) show the same results on an unstructured mixed triangle and quadrilateral grid.

In the second step of the backward trajectory remap method, the preimage of the cell is intersected with the material polygons on the original mesh at time t^n as shown in Fig. 15(b). The areas of each material that lie within the preimage are added to obtain the material contents of the cell. Since the area of the preimage may not exactly match the area of the original cell, the volume fractions are taken with respect to the preimage area. Volume fractions less than 10^{-8} are suppressed.

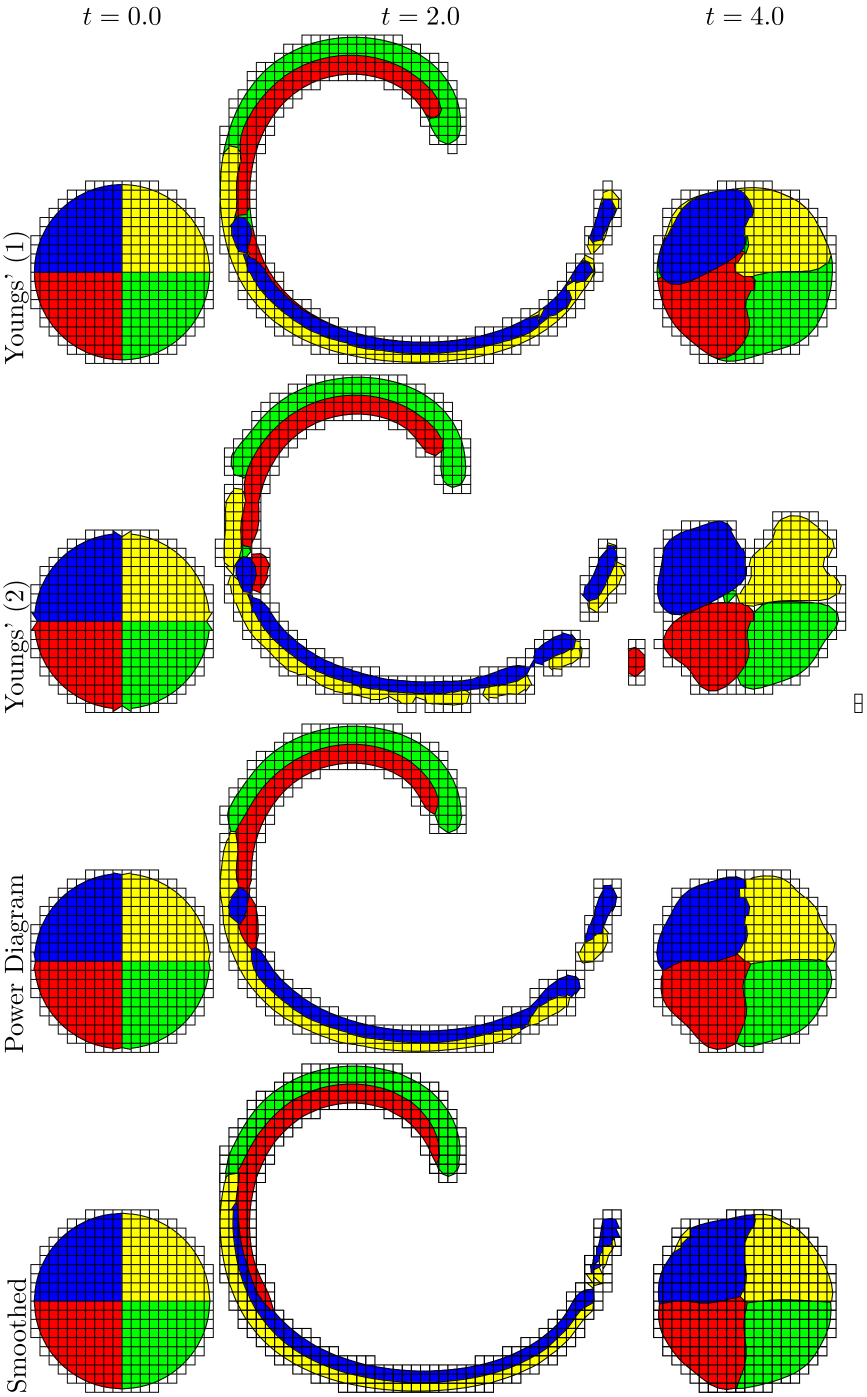

Fig. 17. Material interface configuration for the four material vortex test at maximum stretch time $t = 2.0$ and at complete reversal time $t = 4.0$ run on a 64×64 grid. For the method Youngs' (1), the material ordering was (0, 1, 2, 3, 4). For method Youngs' (2), the order was (1, 2, 3, 4, 0). The initial condition and material numbers are the same as the four material circle shown in Fig. 16 but with the circle center located at (0.5, 0.75).

5.4. Example: diagonal translation of a four material disk

To illustrate the performance of our method in a dynamic example, we choose a simple, five material example that consists of diagonal translation of a four material disk at constant velocity. A four material disk of radius 0.15 is placed at (0.25, 0.25) on a mesh comprising the domain $[0,1] \times [0,1]$. The disk is then translated diagonally with a constant velocity of $\mathbf{u} = (1,1)$ to a final time of $t = 0.5$. While the backward Lagrangian advection scheme has no restrictions on timestep for stability, here the timestep was restricted to emphasize the effect of repeated interface reconstructions. For all simulations, the CFL number, $\nu = \|\mathbf{u}\| \frac{\Delta t}{h}$, was $\frac{1}{8\sqrt{2}}$. The grid spacing, h, is taken as the square root of the area of the smallest grid cell.

In Fig. 16, the final configuration of the circle is shown using different interface reconstruction methods on a 32×32 structured grid and a comparable unstructured, mixed quadrilateral and triangular grid. The material order independent, second-order method (Fig. 16(D1) and (D2)) exhibits no grid artifacts and clearly preserves the structure better than the first order, material order dependent method (Fig. 16(B1),(C1), (B2) and (D2)).

5.5. Example: four material vortex test

A standard volume tracking test case is the vortex consisting of a circle of radius 0.15 centered at (0.5, 0.75) in a $[0,1] \times [0,1]$ domain. To demonstrate material ordering issues, we again use a four material circle. The incompressible velocity field is given by the streamfunction

$$\Psi = \cos\left(\frac{\pi}{4}t\right)\frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi y) \tag{19}$$

with the velocity field defined to be $(u, v) = (-\frac{\partial\Psi}{\partial y}, \frac{\partial\Psi}{\partial x})$. At time $t = 4$, the material configuration should be identical to the initial condition at time $t = 0$.

Here the time step used was $\Delta t = \frac{1}{4}h$ where $h = \frac{1}{64}$ is the mesh spacing and the simulations were run to a final time of $t = 4.0$. The results are shown in Fig. 17. The material order issues with Youngs' method lead to excessive breakup of the interface. On the 64×64 mesh, the solution computed using the second-order method shows problems at the tail of the material. This is due to a poor initial reconstruction and the difficulty in obtaining reliable reference normals for the smoothing procedure when the material has broken up or is very poorly resolved. Overall, our method shows significantly better results than the material order dependent methods. In addition, the power diagram based reconstruction without the smoothing step also shows very good results.

The total change in material volume is defined as

$$e_{vol} = \sum_{m=1}^{N_m=5} \frac{\|V_m^{final} - V_m^0\|}{V_m^0} \tag{20}$$

where V_m^{final} is the total volume of material m after the final time step, and V_m^0 is the total initial volume of material m. For the methods shown, the change in material volume was for Youngs' method with order 01234, $e_{vol} = 2.54e - 3$, for Youngs' method with order 12,340, $e_{vol} = 1.28e - 3$, for the power diagram reconstruction, $e_{vol} = 1.35e - 4$, and for the material order independent second-order method, $e_{vol} = 3.10e - 3$. The volume discrepancy in the backward Lagrangian advection method is largely responsible for the error as all methods exhibit similar errors.

6. Conclusions

We have developed a second-order accurate method for material order independent interface reconstruction of multimaterial cells. It is based on the determination of the relative locations of each material in a cell using linear reconstruction of the volume fraction function, reconstructing the interface using a power diagram and smoothing the interface with respect to its neighbors. The method does not assume a topology for the material regions, i.e. a layer structure or triple point configuration. Furthermore, all of the material regions created are convex which is a requirement for most advection and remapping schemes.

The extension of the smoothing algorithm to three dimensions is challenging. However, extending the material location by linear reconstruction is straightforward as it only relies on computation of the gradient. In addition, the construction of the power diagram to three dimensions only requires the ability to intersect polyhedrons with a halfspace, a procedure detailed in [38]. For the power diagram based reconstruction, the volume fraction matching procedure is dimension independent.

This reconstruction method is currently being implemented in multi-material flow codes to further understand the combined errors of advection and interface reconstruction.

Acknowledgments

The authors would like to thank B.K. Swartz for helpful discussions.

This work was performed under the auspices of the National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396 and supported by the DOE Advanced Simulation and Computing (ASC) program.

References

[1] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Computer Methods in Applied Mechanics and Engineering 99 (1992) 235.

[2] D.J. Benson, Volume of fluid interface reconstruction methods for multi-material problems, Applied Mechanics Review 55 (2) (2002) 151.

[3] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, Journal of Computational Physics 141 (1998) 112.

[4] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annual Review of Fluid Mechanics 31 (1999) 567.

[5] C.W. Hirt, B.D. Nichols, Volume of fluid (vof) method for the dynamics of free boundaries, Journal of Computational Physics 39 (1981) 201.

[6] W.F. Noh, P. Woodward, Slic (simple line interface calculation), in: A.I. van der Vooren, P.J. Zandbergen (Eds.), 5th International Conference on Numerical Methods in Fluid Dynamics, Springer-Verlag, 1976, p. 330.

[7] D.L. Youngs, Time dependent multi-material flow with large fluid distortion, in: K.W. Morton, M.J. Baines (Eds.), Numerical Methods for Fluid Dynamics, Academic Press, New York, 1982, p. 273.

[8] D.L. Youngs, An Interface Tracking Method for a 3d Eulerian Hydrodynamics Code, Technical Report AWE/44/92/35, AWRE Design and Math Division, 1984.

[9] T.J. Barth, Aspects of unstructured grids and finite-volume solvers for Euler and Navier–Stokes equations, in: VKI/NASA/AGARD Special Course on Unstructured Grid Methods for Advection Dominated Flows, vol. R-787, AGARD Publication, 1995.

[10] J.E. Pilliod Jr., E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, Journal of Computational Physics 199 (2004) 465.

[11] S.J. Mosso, B.K. Swartz, D.B. Kothe, R.C. Ferrell, A parallel, volume-tracking algorithm for unstructured meshes, in: P. Schiano, A. Ecer, J. Periaux, N. Satofuka (Eds.), Parallel Computational Fluid Dynamics: Algorithms and Results Using Advanced Computers, Elsevier, 1997, p. 368.

[12] R.V. Garimella, V. Dyadechko, B.K. Swartz, M.J. Shaskov, Interface reconstruction in multi-fluid, multi-phase flow simulations, in: Proceedings of the 14th International Meshing Roundtable, Springer, San Diego, CA, 2005, pp. 19–32. September.

[13] Y. Renardy, M. Renardy, PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method, Journal of Computational Physics 183 (2002) 400.

[14] J.E. Guilkey, T.B. Harman, B. Banerjee, An Eulerian–Lagrangian approach for simulating explosions of energetic devices, Computers and Structures 85 (2007) 660.

[15] B.Y. Choi, M. Bussmann, A piecewise linear approach to volume tracking a triple point, International Journal for Numerical Methods in Fluids 53 (2007) 1005.

[16] V. Dyadechko, M. Shashkov, Multi-material interface reconstruction from the moment data, Journal of Computational Physics (2008), doi:10.1016/j.jcp.2007.12.029.

[17] H.T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, Journal of Computational Physics 226 (2007) 2096.

[18] S.P. Schofield, R.V. Garimella, M.M. Francois, R. Loubère, Material order independent interface reconstruction using power diagrams, International Journal for Numerical Methods in Fluids 56 (2008) 643.

[19] S. Mosso, S. Clancy, A Geometrically Derived Priority System for Young's Interface Reconstruction, Technical Report LA-CP-95-0081, Los Alamos National Laboratory, Los Alamos, NM, 1995.

[20] D.J. Benson, Eulerian finite element methods for the micromechanics of heterogeneous materials: dynamic prioritization of material interfaces, Computer Methods in Applied Mechanics and Engineering 151 (1998) 343–360.

[21] A. Caboussat, M.M. Francois, R. Glowinski, D.B. Kothe, J.M. Sicilian, A numerical method for interface reconstruction of triple points within a volume tracking algorithm. Mathematical and Computer Modelling, in press, doi:10.1016/j.mcm.2008.05.009.

[22] K.S. Bonnell, M.A. Duchaineau, D.R. Schikore, B. Hamann, K.I. Joy, Material interface reconstruction, IEEE Transactions on Visualization and Computer Graphics 9 (4) (2003) 500.

[23] A. Harten, ENO scheme with subcell resolution, Journal of Computational Physics 83 (1989) 148.

[24] B.K. Swartz, The second-order sharpening of blurred smooth borders, Mathematics of Computation 52 (186) (1989) 675.

[25] T.J. Barth, Numerical methods for gasdynamics systems on unstructured grids, in lecture notes in computational science and engineering, "an introduction to recent developments in theory and numerics for conservation laws", in: D. Kroner, M. Ohlberger, C. Rohde (Eds.), Proceedings of the International School on Theory and Numerics for Conservation Laws, Freiburg/Littenweiler, October, 20–24, 1997, Springer, Berlin, 1997, 195–285.

[26] S.F. Bockman, Generalizing the formula for areas of polygons to moments, The American Mathematical Monthly 92 (1989) 131.

[27] L. Margolin, M. Shashkov, Using a curvilinear grid to construct symmetry-preserving discretizations for Lagrangian gas dynamics, Journal of Computational Physics 149 (1999) 389.

[28] F. Aurenhammer, Power diagrams: properties, algorithms and applications, SIAM Journal of Computing 16 (1987) 78.

[29] H. Imai, M. Iri, K. Murota, Voronoi diagram in the Laguerre geometry and its applications, SIAM Journal of Computing 14 (1985) 93.

[30] H. Edelsbrunner, N.R. Shah, Incremental topological flipping works for regular triangulations, Algorithmica 15 (1996) 223.

[31] F. Anton, D. Mioc, On the conversion of ordinary Voronoi diagrams into Laguerre diagrams, in: The Eleventh Canadian Conference on Computational Geometry, Vancouver, Canada, August 1999, pp. 150–153.

[32] S. Fortune, A sweepline algorithms for Voronoi diagrams, Algorithmica 2 (1987) 153.

[33] J. Nocedal, S.J. Wright, Numerical Optimization, Springer, 1999.

[34] J.K. Dukowicz, J.R. Baumgardner, Incremental remapping as a transport/advection algorithm, Journal of Computation Physics 160 (2000) 318.

[35] N. Ashgriz, T. Barbat, G. Wang, A computational Lagrangian–Eulerian advection remap for free surface flows, International Journal for Numerical Methods in Fluids 44 (2004) 1–32.

[36] P. Liovic, M. Rudman, J-L. Liow, D. Lakehal, D. Kothe, A 3D unsplit-advection volume tracking algorithm with planarity-preserving interface reconstruction, Computers and Fluids 35 (2006) 1011.

[37] K. Shahbazi, M. Paraschivoiu, J. Mostaghimi, Second order accurate volume tracking based on remapping for triangular grids, Journal of Computational Physics 188 (2003) 100.

[38] H.T. Ahn, M. Shashkov, Geometric algorithms for 3D interface reconstruction, in: M.L. Brewer, D. Marcum (Eds.), Proceedings of the 16th International Meshing Roundtable, Springer, 2008, pp. 405–422.