

# Moment-of-fluid interface reconstruction

Vadim Dyadechko<sup>\*†</sup> Mikhail Shashkov<sup>†</sup>

LA-UR-05-7571

## Abstract

We designed a new volume-conservative interface reconstruction method. An input data set for the interface reconstruction algorithm consists of volumes and centroids of the cell fractions occupied by different materials. Compared to pure volume fractions, traditionally used by volume-of-fluid (VoF) methods, this choice of the input data allows to define a linear interface approximation per mixed cell (i.e. offers Piecewise-Linear Interface Calculation or Construction (PLIC)) even without exchanging information between neighboring elements. The location of linear interface in each mixed cell is chosen to preserve the volumes and provide the best approximation to the centroids of the cell fractions. The technique proposed yields a second order accurate approximation to interfaces given by simple  $C^2$  curves and is shown to be more accurate than known VoF-PLIC methods.

## 1 Introduction

There is a variety of discrete interface models developed for Eulerian simulations of a multi-phase fluid flow:

- *Level sets* methods [19, 18, 13, 20, 7] work with *implicit* representation of the interface as a zero level set of a *discrete signed distance function* defined for each pair of adjacent fluid phases.

---

<sup>\*</sup>corresponding author; e-mail: vdyadechko@lanl.gov

<sup>†</sup>Mathematical Modeling and Analysis Group (T-7) at Los Alamos National Laboratory

- *Interface tracking* methods use supplementary Lagrangian surface grid that marks the interface between different phases.
- The strategy behind *interface reconstruction* methods is to calculate the interface location at each discrete moment of time from the solution data. Two major groups of interface reconstruction methods should be mentioned here: *volume-of-fluid (VoF)* [4] methods, which construct the interface from the *volumes* of the cell *fractions* occupied by different fluid phases, and *Lagrangian particle* methods [22, 7, 9], which separate the clouds of test particles of different origin.

Either approach has its virtues and drawbacks. The resolution of level set and VoF methods is limited by the resolution of the grid, while interface tracking and particle methods do not have this limitation. Pure particle methods are time consuming and are mostly used in combination with some other major techniques. Interface tracking is inexpensive and straightforward until it comes to interface topology modification, which is a challenging task for tracking algorithms. Level set and interface reconstruction methods, on the contrary, naturally adopt topological changes of the interface.

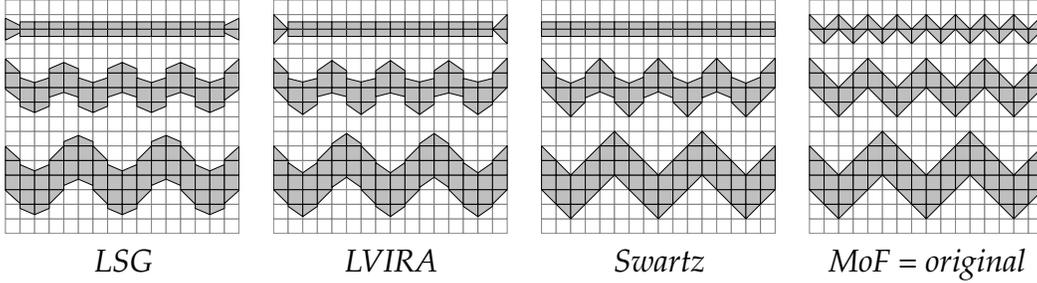
Among all the techniques above, VoF is the only one that preserves the mass of each component. For a wide range of applications the importance of *mass conservation* on a discrete level outbalances all the disadvantages associated with VoF methods.

The basic object of VoF methods is a two-phase medium. Volume fractions of two components sharing a mixed cell are not independent but complement each other to 1. Essentially only the volume fractions  $\alpha$  of single phase, further referred to as a *reference* one, explicitly participate in the interface reconstruction.

The most common interface representation used by VoF methods consists of a single linear interface per mixed cell (so called Piecewise-Linear Interface Calculation or Construction - PLIC). The location of linear interface for a given volume of the reference fraction is uniquely identified by direction of the interface outward (with respect to the reference component) normal. There exists a number of ways to define the normal:

- by estimating the gradient of the *discrete volume fraction function* [23, 24, 2];
- by finding a common linear interface for a cluster of surrounding cells that gives the best approximation to the given volume fractions [14];
- by averaging the normals of linear interfaces that satisfy volume fractions of adjacent mixed cells [6, 8, 21].

*Direct* methods based on estimation of the volume fraction function gradient (Youngs [23, 24], Green-Gauss, and Least Square Gradient (LSG) algorithms [2])



**Figure 1.** Interface approximations obtained with different PLIC algorithms. MoF method is able to reproduce the original shape exactly.

are known to be first order accurate. The *iterative techniques*, like Least square Volume Interface Reconstruction Algorithm (LVIRA) [14] and Mosso-Swartz method [21], are able to approximate  $C^2$  interfaces with second order accuracy. In either case evaluation of the interface outward normal from the volume fractions requires data from the surrounding cells. This inherent feature of VoF methods prohibits the resulting approximation to resolve any interface details smaller than a characteristic size of the cell cluster involved in evaluation of the normal (see Figure 1 on how the quality of LSG, LVIRA, or Swartz interface reconstruction depends on the scale of details in original shape).

In order to overcome this limitation of VoF methods we propose to enrich input data set with *centroids of the cell fractions*. The amount of information carried by the volumes and centroids or, equivalently, by the *first two moments* of the cell fractions is sufficient to define mass-conservative PLIC approximation *even without exchanging data between the cells*. The location of linear interface in each mixed cell is chosen to *preserve the volumes and provide the best approximation to the centroids* of the cell fractions. We call this strategy *moment-of-fluid (MoF) interface reconstruction*.

With no data from adjacent cells participating in evaluation of the interface, the method is able to resolve interface details as small as the cell itself, i.e. two to three times smaller than conventional VoF-PLIC methods (see Figure 1). No data exchange also means that *the grid structure is irrelevant for MoF interface reconstruction*. Second order accurate approximation provided by MoF algorithm guarantees that, once the original interface in a mixed cell is linear, it will be reconstructed *exactly* (Figure 1, MoF).

Compared to alternative approaches, which exploit purely geometrical principles, centroid data involvement has a clear mechanical reason. Each instance of *inexact* interface reconstruction introduces some redistribution of the fluid inside a mixed cell. This fluid motion is unrelated to any physical force presented in

a discrete model. Any displacement  $\Delta\mathbf{x}$  of the component centroid caused by the interface reconstruction can be interpreted as an action of an *artificial* force of magnitude  $\sim m\Delta\mathbf{x}/\Delta t^2$  (here  $m$  is the mass of the component considered, and  $\Delta t$  is the time increment). Therefore by complying with original centroids we *explicitly* reduce these artificial forces and improve approximation properties of the discrete model of fluid dynamics.

The bibliographic search showed that the idea to employ centroid data for VoF methods has been around for a while. Mosso and Clancy [11] proposed to use centroid information to *prioritize materials* in multi-material 2D flow simulations. The same technique was independently presented by Benson [3]. But both these initiatives did not get any further development.

The outline of the rest of this paper is as follows. In Section 2 we describe general concept of moment-of-fluid (MoF) interface reconstruction and address questions related to existence, uniqueness, stability, and accuracy. In Section 3 we present implementation detail of MoF algorithm in 2D. Static tests of interface reconstruction are presented in Section 4. In static tests we start with given interface and computational grid, then we compute the reference moments, perform the interface reconstruction, and then compare the resulting interface with the original one. In this section we present qualitative and quantitative comparison of MoF with other methods.

Dynamic tests are presented in Section 5. In dynamic tests interface is moving with prescribed solenoidal velocity field. Dynamic tests require development of algorithms for advancing in time of volume fractions for VoF-PLIC methods, as well as centroids in case of MoF method. These algorithms are presented in the beginning of Section 5. Next we present numerical results for two test problems using our new MoF method as well as other previously mentioned methods). Finally, in Appendix, we present detail of our implementation of Swartz algorithm.

## 2 Moment-of-fluid interface reconstruction

Let us start with formal definitions. Consider a mixed cell given by an *open polygon*  $\Omega \subset \mathbb{R}^2$ . Each cell fraction occupied by a single component may be represented by a *non-trivial open subset*  $\omega \subset \Omega$  ( $\omega \neq \emptyset$ ,  $\omega \neq \Omega$ ). The *family of all non-trivial open subsets of  $\Omega$*  we denote as  $\mathfrak{S}_\Omega$ . As a non-empty open set,  $\omega \in \mathfrak{S}_\Omega$  always has a positive *volume (area in 2D)*

$$|\omega| \equiv \int_{\omega} d\mathbf{x}, \quad 0 < |\omega| < |\Omega|$$

and a well defined *centroid*

$$\mathbf{x}_c(\omega) \equiv \frac{1}{|\omega|} \int_{\omega} \mathbf{x} \, d\mathbf{x}, \quad \mathbf{x}_c(\omega) \in \text{hull}(\Omega);$$

here  $\text{hull}(\Omega)$  is the *convex hull* of  $\Omega$ .

In practice it is common to specify the volume of  $\omega \in \mathfrak{S}_{\Omega}$  in terms of the *volume fraction*:

$$\mu(\omega) = \frac{|\omega|}{|\Omega|}, \quad 0 < \mu(\omega) < 1.$$

The part of the subset boundary  $\partial\omega$  different from the cell boundary  $\partial\Omega$  represents the *interface*  $\Gamma(\omega)$  between  $\omega$  and its complement  $\Omega \setminus \bar{\omega}$ :

$$\Gamma(\omega) = \partial\omega \setminus \partial\Omega.$$

The cell fractions with *linear interface* form a distinctive class  $\mathfrak{S}_{\Omega}^h$  of *truncation volumes* or *cut-offs*; formally, each cut-off  $\omega_h \in \mathfrak{S}_{\Omega}^h$  is a *non-trivial* intersection

$$\omega_h \equiv \Omega \cap h$$

of  $\Omega$  with an *open half-plane*  $h \subset \mathbb{R}^2$  ( $\omega_h \neq \emptyset$ ,  $\omega_h \neq \Omega$ ).

## 2.1 Problem formulation

Suppose  $\Omega$  contains only two fluid components, and  $\omega^* \in \mathfrak{S}_{\Omega}$  is an original fraction occupied by the reference component. Our optimistic objective is to find a truncation volume  $\omega_h^* \in \mathfrak{S}_{\Omega}^h$  that matches the first two moments of  $\omega^*$  or, equivalently, the *reference volume fraction*  $\mu^* \equiv \mu(\omega^*)$  and the *reference centroid*  $\mathbf{x}^* \equiv \mathbf{x}_c(\omega^*)$ .

The half-plane  $h \subset \mathbb{R}^2$  that defines a truncation volume  $\omega_h \equiv \Omega \cap h$  is uniquely identified with 2 independent parameters: the polar angle of the outward unit normal on  $\partial h$  and the distance of  $\partial h$  from the origin. The reference data space, formed by reference volume fraction and 2 components of reference centroid, is three-dimensional. It is highly unlikely that both *reference moments* can be simultaneously satisfied with a single cut-off: such a problem is *overdetermined*.

Nevertheless one may try to match only a piece of the input data; reference centroid, for example. Even though such a strategy *does not guarantee to preserve the volume* of the reference component, its analysis reveals an important property:

**Property 1** *Each truncation volume is uniquely identified by its centroid.*

This fact implies that the knowledge of centroid location allows to recover truncation volumes (i.e. reproduce linear interfaces) *exactly*.

Since the volume conservation is a must, we have to sacrifice the exact centroid matching. Instead, we are going to *look for a cut-off of the given volume fraction whose centroid provides the best approximation to the reference one*. This strategy, further referred to as a *moment-of-fluid (MoF) interface reconstruction*, makes the object of our study.

Let us introduce the following families of cell fractions:

- all cut-offs of the given volume fraction:

$$\mathfrak{S}_\Omega^{h,\mu^*} = \{ \omega_h \in \mathfrak{S}_\Omega^h \mid \mu(\omega_h) = \mu^* \};$$

- all non-trivial open subsets of the given volume fraction:

$$\mathfrak{S}_\Omega^{\mu^*} = \{ \omega \in \mathfrak{S}_\Omega \mid \mu(\omega) = \mu^* \};$$

and the loci of respective centroids:

$$\mathcal{X}_\Omega^{h,\mu^*} \equiv \mathbf{x}_c(\mathfrak{S}_\Omega^{h,\mu^*}) = \{ \mathbf{x} \in \mathbb{R}^2 \mid \mathbf{x} = \mathbf{x}_c(\omega_h), \omega_h \in \mathfrak{S}_\Omega^{h,\mu^*} \},$$

$$\mathcal{X}_\Omega^{\mu^*} \equiv \mathbf{x}_c(\mathfrak{S}_\Omega^{\mu^*}) = \{ \mathbf{x} \in \mathbb{R}^2 \mid \mathbf{x} = \mathbf{x}_c(\omega), \omega \in \mathfrak{S}_\Omega^{\mu^*} \}.$$

Now moment-of-fluid interface reconstruction can be formally stated as an optimization problem in  $\mathfrak{S}_\Omega^{h,\mu^*}$ :

**Problem 1** Find a cut-off  $\omega_h^* \in \mathfrak{S}_\Omega^{h,\mu^*}$  such that

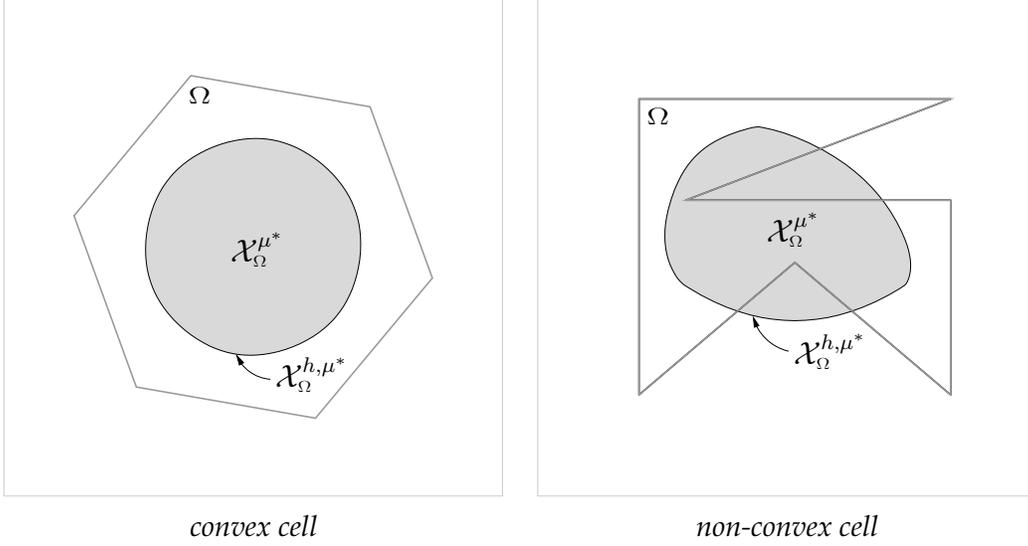
$$\omega_h^* = \arg \min_{\omega_h \in \mathfrak{S}_\Omega^{h,\mu^*}} \| \mathbf{x}_c(\omega_h) - \mathbf{x}^* \|_2^2. \quad (1)$$

Since each truncation volume is uniquely identified by its centroid (Property 1), then Problem 1 is *equivalent* to finding nearest to  $\mathbf{x}^*$  point of  $\mathcal{X}_\Omega^{h,\mu^*}$ :

**Problem 1a** Find a point  $\mathbf{x}_h^* \in \mathcal{X}_\Omega^{h,\mu^*}$  such that

$$\mathbf{x}_h^* = \arg \min_{\mathbf{x}_h \in \mathcal{X}_\Omega^{h,\mu^*}} \| \mathbf{x}_h - \mathbf{x}^* \|_2^2. \quad (1a)$$

Although the latter formulation is of little practical use, it provides a simple model to analyze correctness of MoF interface reconstruction problem: the *existence, uniqueness, and stability* of the solution with respect to reference data perturbation. Since Problem 1a is about approximating  $\mathbf{x}^* \in \mathcal{X}_\Omega^{\mu^*}$  with elements of  $\mathcal{X}_\Omega^{h,\mu^*}$ , our attention should be focused on the properties of these two loci:



**Figure 2.** Examples of  $\mathcal{X}_\Omega^{\mu^*}$  and  $\mathcal{X}_\Omega^{h, \mu^*}$  loci ( $\mu^* = 0.25$ ).

**Property 2**  $\mathcal{X}_\Omega^{\mu^*}$  is a strictly convex closed set with a smooth boundary

$$\partial \mathcal{X}_\Omega^{\mu^*} = \mathcal{X}_\Omega^{h, \mu^*}.$$

This property is illustrated by Figure 2.

**Property 3** Among all the elements of  $\mathfrak{S}_\Omega^{\mu^*}$ , only cut-offs have their centroids located on the boundary of  $\mathcal{X}_\Omega^{\mu^*}$ .

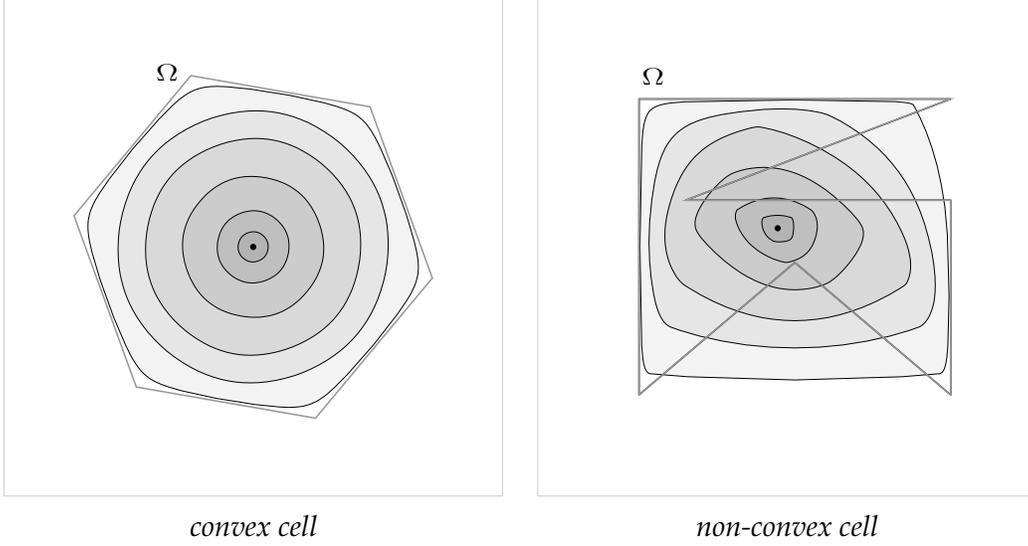
**Property 4** Centroid loci given by different values of volume fraction compose a uniparametric family

$$\{ \mathcal{X}_\Omega^{\mu^*} \mid 0 < \mu^* < 1 \}$$

of nested plane sets:

$$\begin{aligned} \mu_1^* < \mu_2^* &\implies \mathcal{X}_\Omega^{\mu_2^*} \subset \mathcal{X}_\Omega^{\mu_1^*}, \\ \bigcap_{0 < \mu^* < 1} \mathcal{X}_\Omega^{\mu^*} &= \mathbf{x}_c(\Omega), \\ \bigcup_{0 < \mu^* < 1} \mathcal{X}_\Omega^{\mu^*} &= \text{hull}(\Omega). \end{aligned}$$

Figure 3 shows examples of nested centroid loci defined by different values of the reference volume fraction.



**Figure 3.**  $\mathbf{x}_c(\Omega) \in \mathcal{X}_\Omega^{0.90} \subset \mathcal{X}_\Omega^{0.75} \subset \mathcal{X}_\Omega^{0.50} \subset \mathcal{X}_\Omega^{0.25} \subset \mathcal{X}_\Omega^{0.10} \subset \mathcal{X}_\Omega^{0.05} \subset \text{hull}(\Omega)$ .

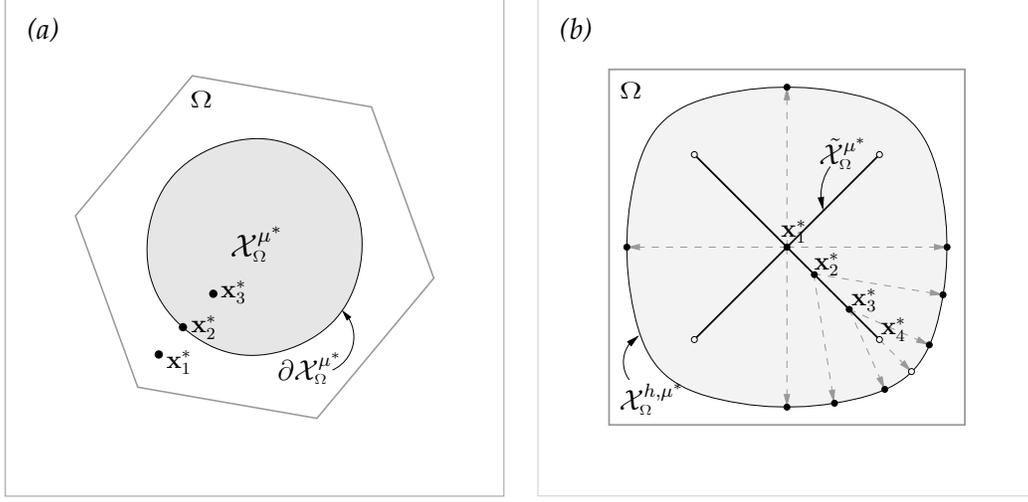
**Existence.** The search space  $(\mathcal{X}_\Omega^{h,\mu^*})$  of continuous objective function  $\|\mathbf{x}_h - \mathbf{x}^*\|_2^2$  is compact (Property 2), therefore by Weierstrass theorem solution of the minimization problem *always exists*.

**Uniqueness.** Even though reference moments are assumed to be given by some cell fraction, i.e.  $\mathbf{x}^* \in \mathcal{X}_\Omega^{\mu^*}$ , one better not count on this fact in a real life calculation: miscellaneous discretization and round-off errors may cause the input data to be inconsistent.

Properties 2 and 3 allow us to classify all input data  $(\mu^*, \mathbf{x}^*) \in ]0, 1[ \times \mathbb{R}^2$  into three categories (see Figure 4a):

- 1)  $\mathbf{x}^* \notin \mathcal{X}_\Omega^{\mu^*}$ : the reference moment data are inconsistent, i.e. there is no cell fraction that satisfies them. Some consolation is supplied by the fact that solution of (1) (which is *unique* because  $\mathbf{x}^*$  is located outside the convex region bounded by  $\mathcal{X}_\Omega^{h,\mu^*} \equiv \partial\mathcal{X}_\Omega^{\mu^*}$ ) gives *the best* approximation to the reference centroid *among all the subsets of the given volume fraction*.
- 2)  $\mathbf{x}^* \in \partial\mathcal{X}_\Omega^{\mu^*}$ : the source of the reference data is known to be a truncation volume (Property 3), which can be *uniquely* (Property 1) identified by solving (1).
- 3)  $\mathbf{x}^* \in \mathcal{X}_\Omega^{\mu^*} \setminus \partial\mathcal{X}_\Omega^{\mu^*}$ : the original cell fraction  $\omega^*$  is *not* a cut-off. This type of the input is the most common and also requires more careful analysis.

To start with, Problem 1a in this case may have multiple solutions. The



**Figure 4.** (a) Classification of the reference data. (b) Example of  $\tilde{\mathcal{X}}_{\Omega}^{\mu^*}$  for square cell ( $\mu^* = 0.1$ ). We also show here reference centroid samples  $\mathbf{x}_1^*, \mathbf{x}_2^*, \mathbf{x}_3^* \in \tilde{\mathcal{X}}_{\Omega}^{\mu^*}$ ,  $\mathbf{x}_4^* \notin \tilde{\mathcal{X}}_{\Omega}^{\mu^*}$  and their respective solutions on  $\mathcal{X}_{\Omega}^{h, \mu^*}$ .

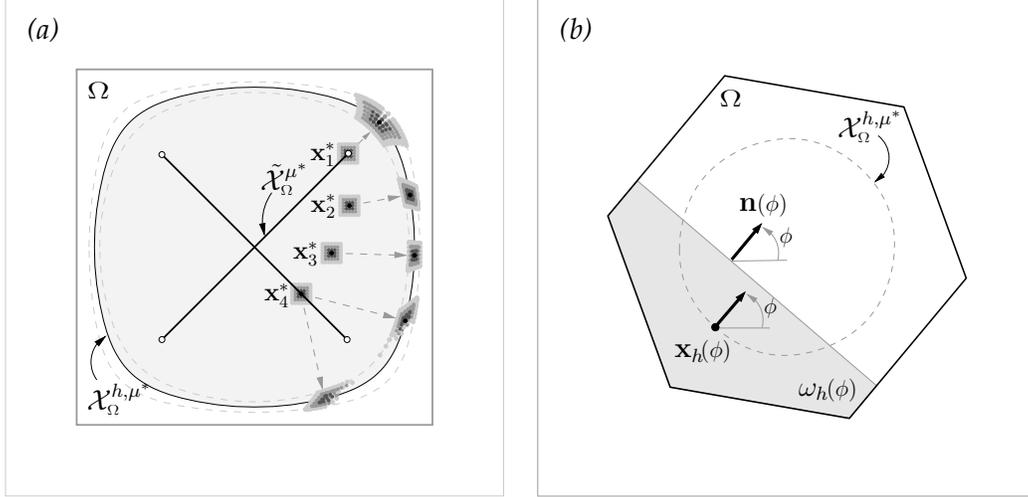
number of multiple solutions is always finite and can be as big as the number of vertices of the polygon given by the convex hull of a cell.

We use symbol  $\tilde{\mathcal{X}}_{\Omega}^{\mu^*}$  to denote the *locus of all reference centroids defining multiple solutions of (1a)*.  $\tilde{\mathcal{X}}_{\Omega}^{\mu^*}$  is a non-empty connected set composed of a finite number of smooth segments that form a tree-like pattern with open tip branches (see Figure 4b). Since  $\tilde{\mathcal{X}}_{\Omega}^{\mu^*}$  has zero area, one can consider the input  $\mathbf{x}^* \in \tilde{\mathcal{X}}_{\Omega}^{\mu^*}$  to be *improbable*. Therefore *MoF interface reconstruction is unique with absolute certainty*.

The better  $\Gamma(\omega^*)$  can be approximated with a line segment, the closer to  $\partial\mathcal{X}_{\Omega}^{\mu^*}$  the reference centroid is. This observation is very useful in the light of the fact that the lower bound for the distance between  $\tilde{\mathcal{X}}_{\Omega}^{\mu^*}$  and  $\partial\mathcal{X}_{\Omega}^{\mu^*}$  is strictly positive for any particular value of  $\mu^*$ . More detailed analysis shows that *for any original interface with sufficiently low (compared to  $1/\text{diam}(\Omega)$ ) curvature the reference centroid is located beyond the reach of  $\tilde{\mathcal{X}}_{\Omega}^{\mu^*}$ , i.e. the MoF interface reconstruction is unique*.

**Stability.** Each input  $(\mu^*, \mathbf{x}^*)$  of MoF interface reconstruction problem can be classified either as *unique* or *multiple*, depending on the number of resulting solutions. If  $\mathcal{M}_{\Omega}$  is a subset of the *input data space*  $\mathcal{I}_{\Omega} \equiv ]0, 1[ \times \mathbb{R}^2$  composed of all multiple inputs

$$\mathcal{M}_{\Omega} = \{ (\mu^*, \mathbf{x}^*) \in \mathcal{I}_{\Omega} \mid \mathbf{x}^* \in \tilde{\mathcal{X}}_{\Omega}^{\mu^*} \},$$



**Figure 5.** (a) Sample solutions diffused by the fuzziness of the reference data (of both centroid and volume fraction). (b) The parametrization of the search space.

then the complement of  $\mathcal{M}_\Omega$

$$\mathcal{I}_\Omega \setminus \mathcal{M}_\Omega = \{ (\mu^*, \mathbf{x}^*) \in \mathcal{I}_\Omega \mid \mathbf{x}^* \notin \tilde{\mathcal{X}}_\Omega^{\mu^*} \}$$

is composed of all unique inputs. Not all unique inputs are stable: any input that corresponds to the tip of an  $\tilde{\mathcal{X}}_\Omega^{\mu^*}$  branch (sample  $\mathbf{x}_4^*$  on Figure 4b), under infinitesimally small perturbation may turn into a multiple input. The stability zone of MoF interface reconstruction is given by the complement of the  $\mathcal{M}_\Omega$  closure:

$$\mathcal{I}_\Omega \setminus \overline{\mathcal{M}_\Omega} = \{ (\mu^*, \mathbf{x}^*) \in \mathcal{I}_\Omega \mid \mathbf{x}^* \notin \overline{\tilde{\mathcal{X}}_\Omega^{\mu^*}} \}.$$

Any sufficiently small perturbation of  $(\mu^*, \mathbf{x}^*) \in (\mathcal{I}_\Omega \setminus \overline{\mathcal{M}_\Omega})$  defines a unique input  $(\mu^* + \delta\mu^*, \mathbf{x}^* + \delta\mathbf{x}^*)$  with respective solution close to the unperturbed one (see fuzzy reference data samples  $\mathbf{x}_2^*$ ,  $\mathbf{x}_3^*$  and their respective solutions on Figure 5a).

**Approximation properties.** MoF interface reconstruction not only preserves volumes of cell fractions but also results in *minimal defect of the first moment*

$$\Delta M_1 \equiv \|\mathbf{M}_1(\omega_h^*) - \mathbf{M}_1(\omega^*)\|_2 = |\Omega| \mu^* \|\mathbf{x}_h^* - \mathbf{x}^*\|_2$$

attainable with a PLIC approximation; here  $\mathbf{M}_1(\omega) \equiv \int_\omega \mathbf{x} \, dx$  is the first moment of  $\omega \in \mathfrak{S}_\Omega$ .

Another way to quantify the interface reconstruction error is to measure the area of the *symmetric difference* between  $\omega_h^*$  and  $\omega^*$  :

$$\Delta\omega \equiv |\omega_h^* \Delta \omega^*| = |(\omega_h^* \cup \omega^*) \setminus (\omega_h^* \cap \omega^*)|.$$

One may also find the maximum deviation of the original interface  $\Gamma(\omega^*)$  from the cut-line  $\partial h^*$  defining  $\Gamma(\omega_h^*) = \Omega \cap \partial h^*$  :

$$\Delta\Gamma \equiv \text{dist}(\Gamma(\omega_h^*), \partial h^*) = \max_{\mathbf{x} \in \Gamma(\omega_h^*)} \min_{\mathbf{y} \in \partial h^*} \|\mathbf{x} - \mathbf{y}\|_2,$$

to be more informative than  $\Delta M_1$  or  $\Delta\omega$ .

As long as the original interface  $\Gamma(\omega^*)$  is known to be a  $C^2$  curve with curvature radius bounded from below by a positive constant  $R$ , the following estimates hold true:

$$\Delta M_1 = O(d^5/R^2), \quad \Delta\omega = O(d^3/R), \quad \Delta\Gamma = O(d^2/R), \quad (2)$$

where  $d = \text{diam}(\Omega)$ .  $\Delta\Gamma$  estimate shows that *MoF interface reconstruction is second order accurate*.

For a non-smooth interface all errors above reach their respective pessimistic upper bounds:

$$\Delta M_1 = O(d^3), \quad \Delta\omega = O(d^2), \quad \Delta\Gamma = O(d). \quad (3)$$

### 3 Implementation of MoF Algorithm in 2D

Let  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^2$  be the vertices of a mixed cell  $\Omega$  enumerated in *counter-clockwise order*. The area and centroid of  $\Omega$  (as well as of any other polygon with  $n$  vertices) can be calculated by formulas:

$$|\Omega| = \frac{1}{2} \sum_{i=1}^n [\mathbf{x}_i \times \mathbf{x}_{i+1}], \quad (4)$$

$$\mathbf{x}_c(\Omega) = \frac{1}{6|\Omega|} \sum_{i=1}^n [\mathbf{x}_i \times \mathbf{x}_{i+1}] (\mathbf{x}_i + \mathbf{x}_{i+1}), \quad (5)$$

where  $[\cdot \times \cdot]$  is *2D vector product*, and  $\mathbf{x}_{n+1} \equiv \mathbf{x}_1$ .

In order to proceed with numerical optimization, one have to introduce a suitable parametrization of the search space. Since  $\mathcal{X}_\Omega^{h,\mu^*}$  is a smooth boundary of a strictly convex set, each  $\mathbf{x}_h \in \mathcal{X}_\Omega^{h,\mu^*}$  can be uniquely identified with the polar

angle  $\phi$  of the local inward normal on  $\mathcal{X}_\Omega^{h,\mu^*}$  (Figure 5b). Note that the same polar angle specifies the interface outward normal for respective cut-off  $\omega_h \in \mathfrak{S}_\Omega^{h,\mu^*}$ ,  $\mathbf{x}_c(\omega_h) = \mathbf{x}_h$ . There is no need to know  $\mathbf{x}_h(\phi)$  dependence explicitly to evaluate the objective function

$$f(\phi) \equiv \|\mathbf{x}_h(\phi) - \mathbf{x}^*\|_2^2; \quad (6)$$

given  $\phi \in \mathbb{R} \bmod 2\pi$ , one can

- 1) calculate unit vector  $\mathbf{n}(\phi) \equiv (\cos \phi, \sin \phi)$ ;
- 2) find the truncation volume  $\omega_h(\phi) \in \mathfrak{S}_\Omega^{h,\mu^*}$  with the *interface outward unit normal*  $\mathbf{n}(\Gamma(\omega_h(\phi))) = \mathbf{n}(\phi)$ ;
- 3) calculate  $\mathbf{x}_h(\phi)$  as the centroid of  $\omega_h(\phi)$  using (5).

Instead of finding  $\omega_h \in \mathfrak{S}_\Omega^{h,\mu^*}$  with  $\mathbf{n}(\Gamma(\omega_h)) = \mathbf{n}(\phi)$ , it is more convenient to identify a cut-off of the given volume fraction among the *truncation volumes with prescribed interface normal*

$$\mathfrak{S}_\Omega^{h,\mathbf{n}(\phi)} = \{ \omega_h \in \mathfrak{S}_\Omega^h \mid \mathbf{n}(\Gamma(\omega_h)) = \mathbf{n}(\phi) \}.$$

Each truncation volume  $\omega_h \in \mathfrak{S}_\Omega^{h,\mathbf{n}(\phi)}$  is described by its height  $\xi$  (the ‘‘vertical’’ direction is assumed to be given by  $\mathbf{n}(\phi)$ ). Cut-off volume fraction  $\mu(\xi) \equiv \mu(\omega_h(\xi))$ ,  $\omega_h \in \mathfrak{S}_\Omega^{h,\mathbf{n}(\phi)}$  is a *continuous monotone* function of  $\xi$  (Figure 6), i.e. has a well define inverse. Therefore the height  $\xi^*$  of  $\omega_h(\phi) \in \mathfrak{S}_\Omega^{h,\mu^*} \cap \mathfrak{S}_\Omega^{h,\mathbf{n}(\phi)}$  can be found by solving

$$\mu(\xi^*) = \mu^*. \quad (7)$$

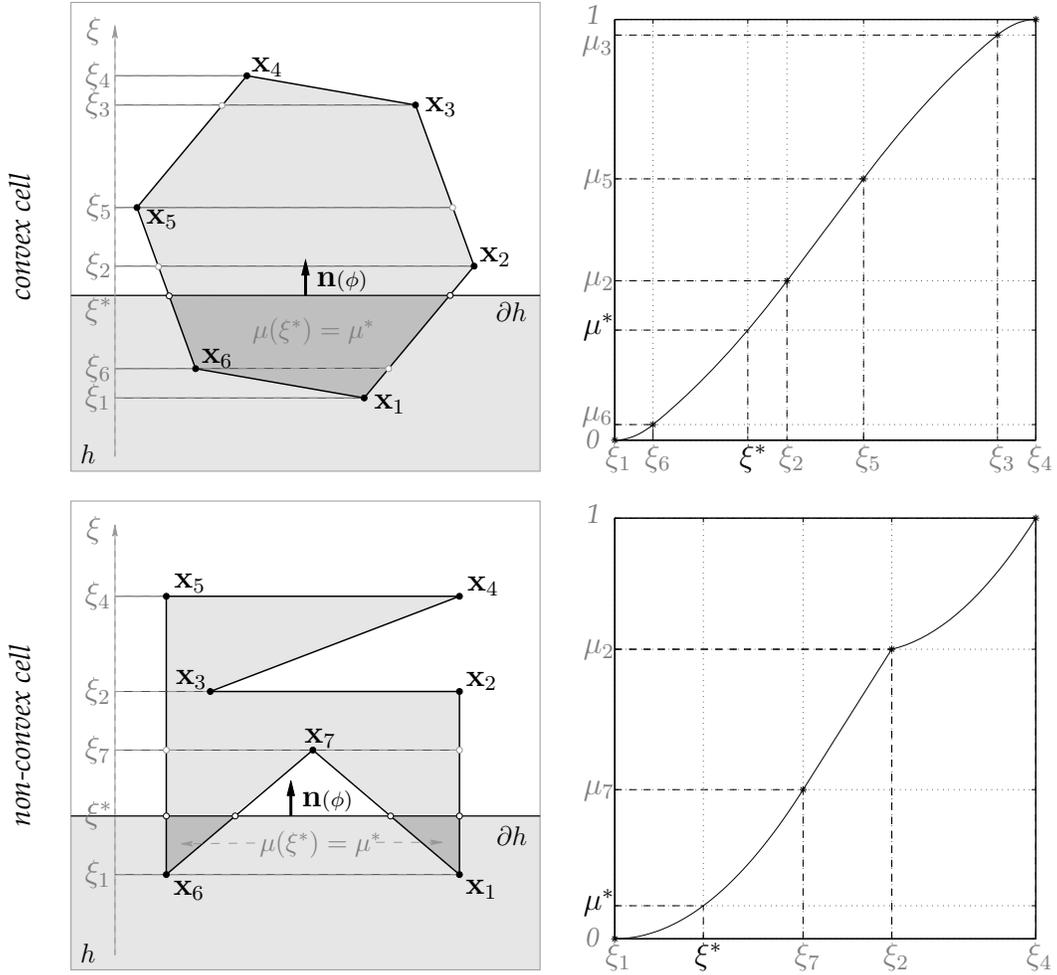
Second derivative of  $\mu(\xi)$  is a *piecewise-constant* function with discontinuity points given by the vertex altitudes  $\xi_1, \xi_2, \dots, \xi_n$ . This fact along with monotonicity of  $\mu(\xi)$  yields an efficient solution strategy for (7), which requires only  $O(n)$  operations for a convex  $\Omega$  and  $O(n^2)$  operations in general case. We call it *flood algorithm*, since it models the process of filling a vessel with fluid.

**Flood algorithm.** Let  $(i_1, i_2, \dots, i_n)$  be an index permutation that puts vertex altitudes in a non-descending order:

$$0 \equiv \xi_{i_1} \leq \xi_{i_2} \leq \dots \leq \xi_{i_n}. \quad (8)$$

Since  $\mu(\xi)$  is either quadratic or linear on each interval  $[\xi_{i_k}, \xi_{i_{k+1}}]$ ,  $k = \overline{1, n-1}$ , then the area of  $\Omega$ , enclosed between two levels  $\xi = \xi_{i_k}$  and  $\xi = \xi_{i_{k+1}}$ , can be calculated by the *trapezoid rule*:

$$|\Omega| (\mu_{k+1} - \mu_k) = \frac{1}{2} (\xi_{i_{k+1}} - \xi_{i_k}) (|\Gamma_{k+1}| + |\Gamma_k|). \quad (9)$$



**Figure 6.** Volume fraction  $\mu(\omega_h)$ ,  $\omega_h \in \mathfrak{S}_\Omega^{h, \mathbf{n}(\phi)}$  as a function the cut-off height  $\xi$ .

Here  $|\Omega|$  is the area of the cell,  $\mu_k \equiv \mu(\xi_{i_k})$ , and  $|\Gamma_k|$  is the total length of  $\Gamma_k \equiv \Gamma(\xi_{i_k} \equiv \Gamma(\omega_h(\xi_{i_k})))$ ,  $k = \overline{1, n}$ .

First one have to find the interval  $[\xi_{i_{k^*}}, \xi_{i_{k^*+1}}]$  of quadratically (linearity) of  $\mu(\xi)$  that includes  $\mu^*$  :

$$\mu_{i_{k^*}} < \mu^* \leq \mu_{i_{k^*+1}}, \quad 1 \leq k^* \leq n-1. \quad (10)$$

Expression (9) gives a basis for recurrent calculation of all  $\mu_k$ ,  $k = \overline{2, n-1}$ . The search starts from the "bottom"

$$\mu_1 = 0, \quad |\Gamma_1| = 0,$$

and goes all the way "up"

$$\mu_k = \mu_{k-1} + \frac{1}{2|\Omega|} (\xi_{i_k} - \xi_{i_{k-1}}) (|\Gamma_k| + |\Gamma_{k-1}|), \quad k = 2, 3, \dots, \quad (11)$$

until the reference volume fraction is framed.

Note that each step of the iterative process (11) involves identification of respective "cross section"  $\Gamma_k$ ,  $k = \overline{2, n-1}$  ( $O(1)$  operations for a convex cell,  $O(n)$  otherwise).

Once the interval of interest (10) is known, solution of (7) can be found by means of interpolation:

- if  $\mu(\xi)$  is linear on  $[\xi_{i_{k^*}}, \xi_{i_{k^*+1}}]$ , i.e. the framed area is a parallelogram ( $|\Gamma_{k^*+1}| = |\Gamma_{k^*}|$ ), then

$$\xi^* = \xi_{i_{k^*}} + \frac{\mu^* - \mu_{k^*}}{\mu_{k^*+1} - \mu_{k^*}} (\xi_{i_{k^*+1}} - \xi_{i_{k^*}});$$

- otherwise  $\mu(\xi)$  is quadratic and

$$\xi^* = \sqrt{\xi_{i_{k^*}}^2 + \frac{\mu^* - \mu_{k^*}}{\mu_{k^*+1} - \mu_{k^*}} (\xi_{i_{k^*+1}}^2 - \xi_{i_{k^*}}^2)}.$$

Bracketing the reference volume fraction (11) is the most expensive part of the whole algorithm (the complexity of the initial vertex altitude sort (8) is just  $O(n)$  for a convex cell and  $O(n \log n)$  in general case). One can reduce the expected bracketing time by half by implementing the search (11) in backward order for all  $\mu^* > 1/2$ . In this case the search starts from the "top" ( $\mu_n = 0$ ,  $|\Gamma_n| = 0$ ) and goes all the way "down" ( $k = n-1, n-2, \dots$ ) to the interval of interest. Another reason to implement the backward search is more accurate evaluation of  $\xi^*$  for  $\mu^* \approx 1$ .

**Objective function derivative.** The choice of the polar angle as an independent variable yields inexpensive evaluation of the first derivative of  $\mathbf{x}_h(\phi)$ :

- whenever the interface  $\Gamma(\phi) \equiv \Gamma(\omega_h(\phi))$  consists of a single segment (which is always the case for a convex cell),

$$\mathbf{x}'_h(\phi) = \frac{1}{12} \frac{|\Gamma(\phi)|^3}{\mu^*|\Omega|} \mathbf{t}(\phi), \quad (12)$$

where  $\mathbf{t}(\phi) = [-\sin \phi, \cos \phi]^T$  is a *counter-clockwise unit tangent* on  $\Gamma(\phi)$ .

- for a non-convex cell the interface  $\Gamma(\phi)$  may consist of several separate segments of the cut-line (see Figure 6, non-convex cell); in this case

$$\mathbf{x}'_h(\phi) = \frac{M_2(\Gamma(\phi))}{\mu^*|\Omega|} \mathbf{t}(\phi), \quad (13)$$

where  $M_2(\Gamma(\phi))$  is a *second central moment* of the interface  $\Gamma(\phi)$ :

$$M_2(\Gamma) = \int_{\Gamma} \|\mathbf{x} - \mathbf{x}_h(\Gamma)\|_2^2 d\Gamma, \quad \mathbf{x}_h(\Gamma) = \frac{1}{|\Gamma|} \int_{\Gamma} \mathbf{x} d\Gamma.$$

Expressions (12) and (13) complete the *chain rule* for the first derivative of the objective function

$$f'(\phi) = 2 \left( (\mathbf{x}_h(\phi) - \mathbf{x}^*) \cdot \mathbf{x}'_h(\phi) \right).$$

Here  $(\cdot \cdot \cdot)$  is a *dot product*.

**Numerical optimization** of (6) can be performed with any *line search* algorithm [12, 5, 15] available. One has to be careful choosing the *initial guess* and *trial step selection strategy* for the line search algorithm: even when the objective function (6) has a unique *global* minimum, it may still possess some other *local* minima (see Figure 7,  $\mu^* = 0.02$ ), which can distract the algorithm.

A safe initial guess is provided by the polar angle of  $\mathbf{x}_c(\Omega) - \mathbf{x}^*$  vector. If  $\mathcal{X}_{\Omega}^{h,\mu^*}$  were a circle, this choice of  $\phi_0$  would give us the exact minimizer of (6). The more isotropic the shape a mixed cell is, the better  $\mathcal{X}_{\Omega}^{h,\mu^*}$  can be approximated with a circle for a moderate value of  $\mu^*$ , the closer to the minimizer of (6) the initial guess is.

We also claim that suggested  $\phi_0$  either specifies the point on the slope of a global minimum valley or marks the top of the hill that divides two global minima. Therefore a conservative trial step selection strategy is guaranteed to lead the line search started at  $\phi_0$  to a global minimum. The line search is terminated when the polar angle of the normal stabilizes within  $tol_{\phi} = 10^{-6}$ .

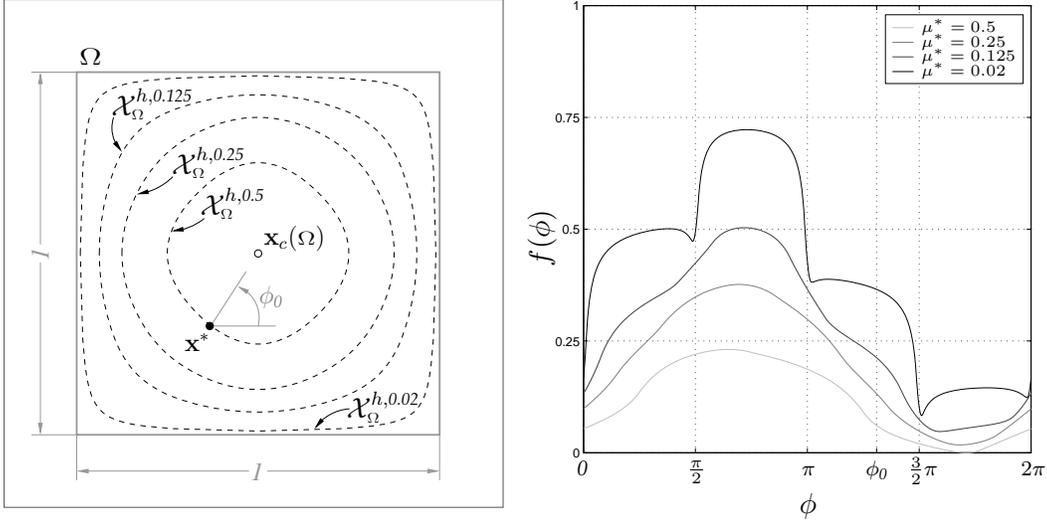


Figure 7. Sample objective function graphs.

## 4 Static tests

The idea of static test for an interface reconstruction algorithm is trivial: given a particular shape of the reference component and a computational grid, one has to calculate the reference moments, run the interface reconstruction, and then compare the resulting shape of the original one.

Along with MoF method we tested three VoF-PLIC methods for arbitrary polygonal grids:

- Least Square Gradient (LSG) with reciprocal quadratic distance weights[2], which we refer to as LSGQ,
- LVIRA [14],
- Swartz method<sup>1)</sup> [21, 10].

The choice of the last two methods is pretty obvious: these are the only original second order accurate VoF-PLIC methods suitable for unstructured grids. Both LVIRA and Swartz algorithms are *iterative* and rely on a *reasonable initial guess*, usually provided by a *direct* VoF-PLIC method (LSGQ in our case).

The first order accurate algorithms are represented by LSG; all one have to know about the grid structure to run it a list of neighbors for each mixed cell. Frankly, the only real alternative to LSG is Green-Gauss algorithm [2], but the latter requires adjacent cells to be properly ordered. On a uniform rectangular

<sup>1)</sup> The implementation details can be found in Appendix.

grid both LSGQ and Green-Gauss give *the same* result as Youngs method [24] with optimal parameter.

The results produced by these PLIC algorithms for six different shapes are presented on Figures 8 and 9. Although original shapes are not shown, their names are self-explanatory. As one can see, MoF method has higher resolution than VoF alternatives, adds no blurring, and takes the maximum out of PLIC approximation: polygonal shapes “zigzags” and “compass rose”, which are specially designed to have a piecewise-linear boundary that can be captured on the  $16 \times 16$  grid with a single linear segment per cell, are reproduced by MoF algorithm exactly.

Our next step is quantification of the interface reconstruction error. With focus of discussion shifting from a single mixed cell to the entire computational grid, it is necessarily to change the notation. Let  $\Omega \subset \mathbb{R}^2$  now represent a polygonal domain *quasi-uniformly* partitioned into  $N$  polygonal cells  $\{\Omega_i\}_{i=1}^N$  :

$$\overline{\Omega} = \bigcup_{i=1}^N \overline{\Omega}_i, \quad \Omega_i \cap \Omega_j = \emptyset \quad \forall i \neq j;$$

open  $\omega^* \subset \Omega$  and  $\omega_h^* \subset \Omega$  represent the original and the reconstructed shapes of the reference component. The restrictions of  $\omega^*$  and  $\omega_h^*$  to a single cell  $\Omega_i$ ,  $i = 1, N$  will be referred to as  $\omega_i^*$  and  $\omega_{h,i}^*$  respectively.

Several ways to quantify the interface reconstruction error were mentioned in Subsection 2.1. Since we already had in our possession fast polygon intersection routine<sup>2)</sup>, we decided to measure

$$\Delta\omega = |\omega^* \Delta \omega_h^*| = \sum_{i=1}^N |\omega_{h,i}^* \Delta \omega_i^*|.$$

The interface reconstruction algorithms we examine are volume-conservative, i.e.  $|\omega_{h,i}^*| = |\omega_i^*|$ ,  $i = \overline{1, N}$ , therefore

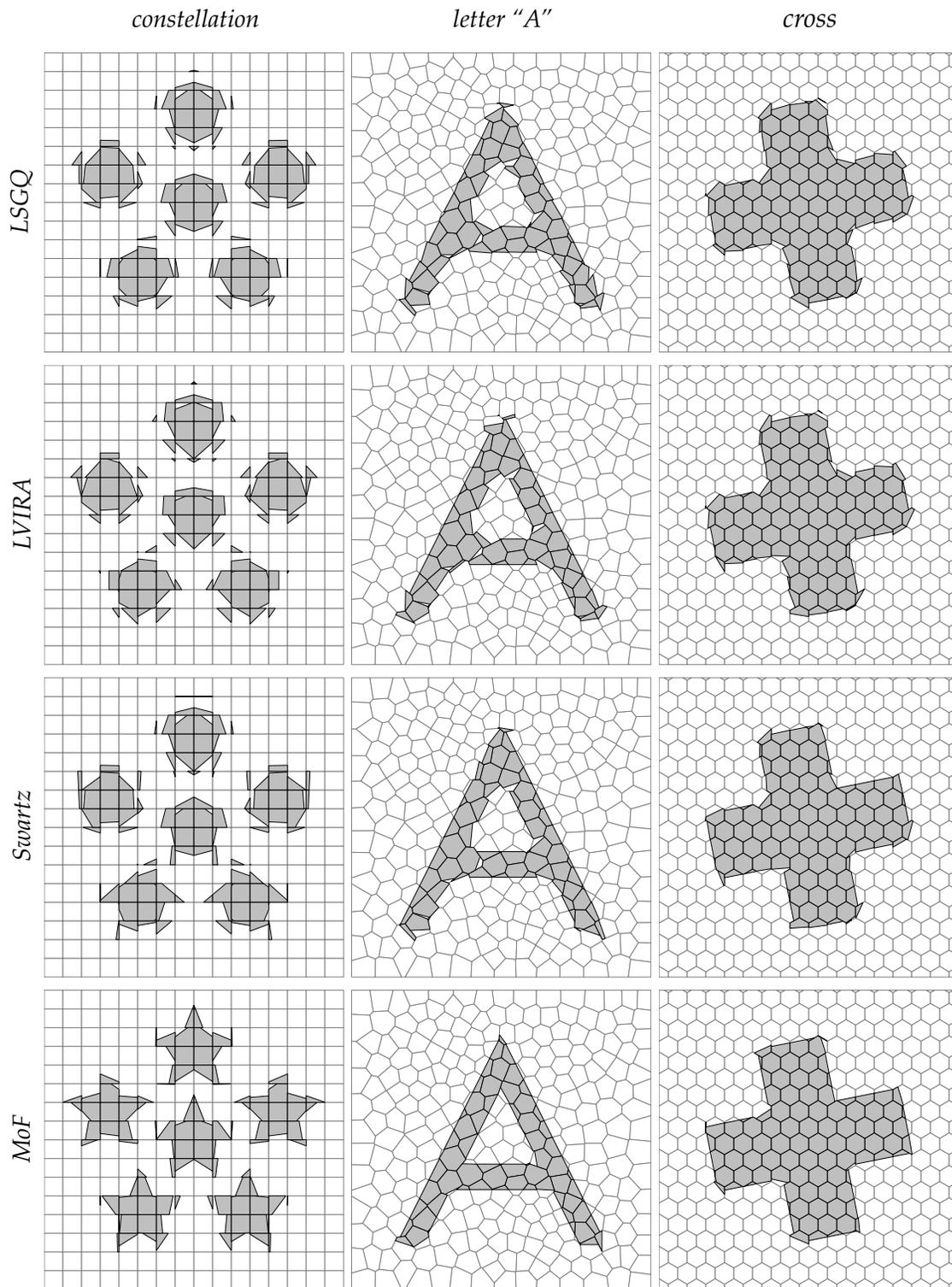
$$|\omega_{h,i}^* \Delta \omega_i^*| = \begin{cases} 2(|\omega_i^*| - |\omega_i^* \cap \omega_{h,i}^*|), & \text{if } \Omega_i \text{ is a mixed cell,} \\ 0, & \text{otherwise,} \end{cases}$$

which yields the following practical expression for  $\Delta\omega$  :

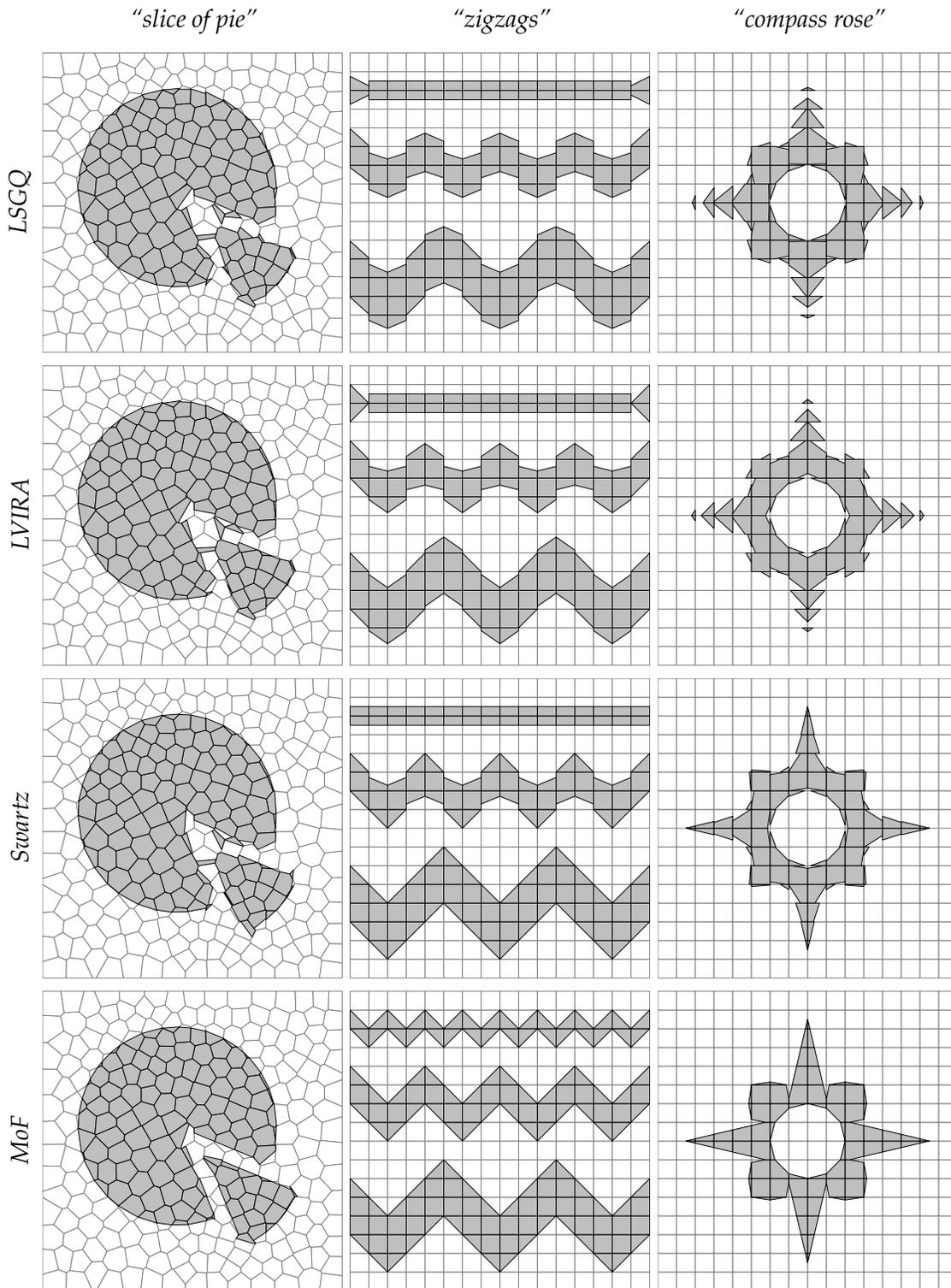
$$\Delta\omega = 2 \sum_{i=1}^M (|\omega_i^*| - |\omega_i^* \cap \omega_{h,i}^*|); \quad (14)$$

---

<sup>2)</sup> Due to COnservative REmapper (CORE) library by M. Staley [17].



**Figure 8.** Results of the interface reconstruction with various PLIC techniques.



**Figure 9.** More interface reconstruction results.

here  $M$  is the total number of mixed cells, without loss of generality we can assume that all mixed cells are enumerated from 1 to  $M$ .

This area, divided by the length of the original interface  $|\partial\omega^*|$ , gives the *average distance between the reconstructed and original interfaces*. To make this quantity scale-independent, we normalize it by a *characteristic size*  $\tilde{L}$  of the original shape  $\omega^*$ ; the result is referred to as a *normalized average deviation*:

$$\tilde{\epsilon}_{avg} = \frac{1}{\tilde{L}} \frac{|\omega^* \Delta \omega_h^*|}{|\partial\omega^*|}.$$

Another important error characteristics we can get with  $\{|\omega_i^* \cap \omega_{h,i}^*|\}_{i=1}^M$  is the *maximum of the normalized average deviation per (mixed) cell*:

$$\tilde{\epsilon}_{max} = \max_{i=1, M} \frac{1}{\tilde{L}} \frac{|\omega_i^* \Delta \omega_{h,i}^*|}{|\Gamma(\omega_{h,i}^*)|},$$

which indicates the worst local error attainable.

To figure out how these errors scale with mesh refinement we conducted a series of direct error measurements on a sequence of uniform rectangular grids ( $\Omega = ]0, 1[ \times ]0, 1[$ ) with mesh spacing  $h$  varying from  $1/2$  to  $1/2^{10}$  ( $h_k = 1/2^k$ ,  $k = 1, 10$ ).

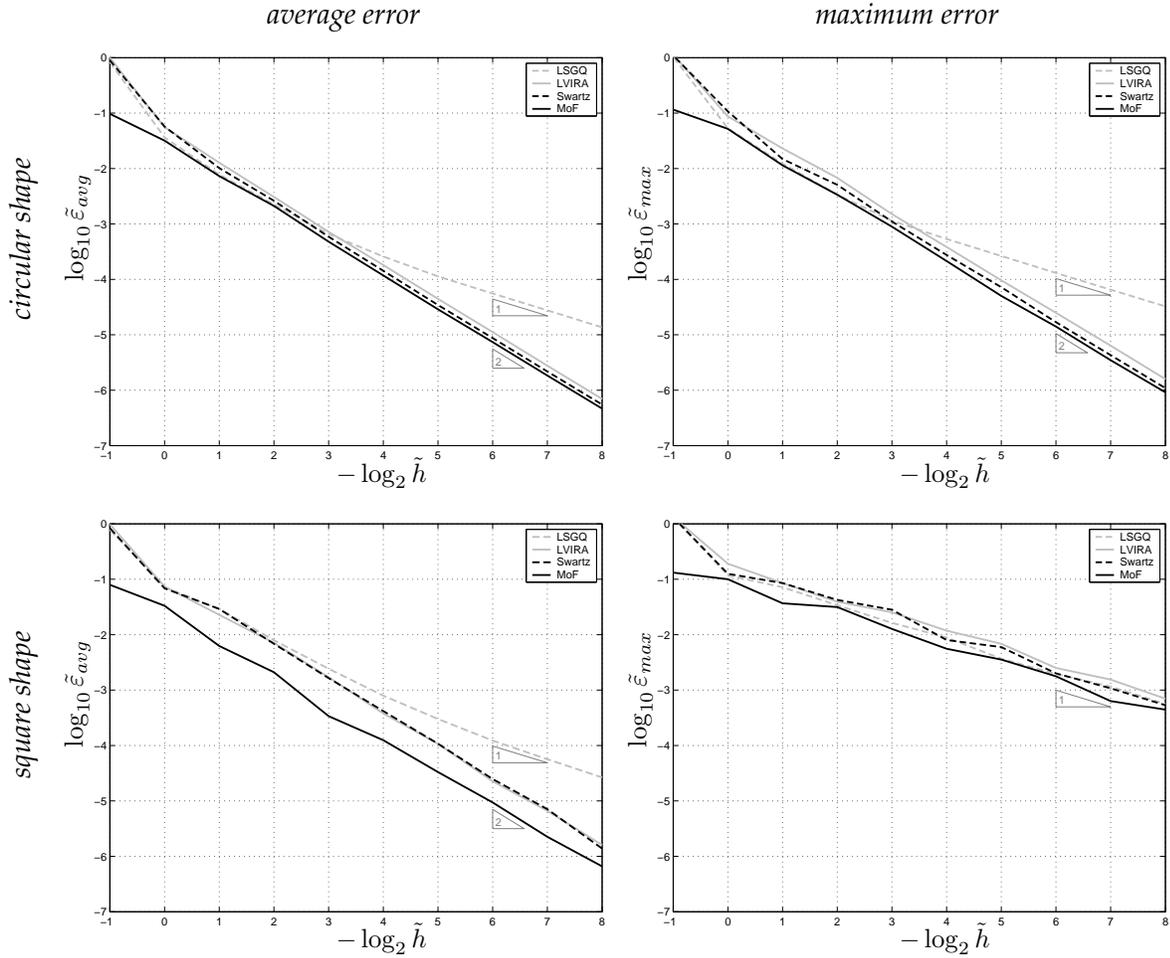
Each PLIC algorithm was tested against two shapes:

- 1) circle of radius  $\tilde{L} = 0.25$  centered at  $(0.5 + 1/17, 0.5 + 1/41)$ ;
- 2)  $2\tilde{L} \times 2\tilde{L} = 0.5 \times 0.5$  square centered at  $(0.5 + 1/17, 0.5 + 1/41)$  and rotated counter-clockwise by  $\pi/16$  radians;

Both shapes have the same characteristic scale  $\tilde{L} = 0.25$ , the position and the orientation were chosen to eliminate the influence of the symmetry factor on error readings.

The results of these tests are put on graphs (Figure 10). Here error is related to the normalized mesh spacing  $\tilde{h} = h/\tilde{L}$ . Both horizontal and vertical axes have logarithmic scale.

Let us first comment the circle test results. As one might have anticipated, LVIRA, Swartz and MoF algorithms provide second order accurate approximation to the  $C^2$ -smooth interface, while LSGQ is only first order accurate. Uniform curvature of the interface explains similar behavior of both average and maximum errors, although average error is somewhat lower and less volatile due to the aggregate nature of this indicator. MoF algorithm results in the lowest error. Average errors of LVIRA and Swartz algorithms are 50+% and 18+% higher than respective MoF error. Even though LSGQ is asymptotically less accurate than LVIRA and Swartz, the former algorithm exhibits significantly smaller error (just 6% above the MoF) on coarse and medium scales and starts to loose the



**Figure 10.** Interface reconstruction error.

advantage only after the cell size decreases below  $1/8$  of the interface curvature radius.

Original interface in the second test is not smooth, and that is why all four methods demonstrate local error of order  $O(h)$ . Asymptotic average error of LSGQ is also  $O(h)$ , while for all other three methods only  $O(h^2)$ . This behavior has a clear explanation: all second order accurate methods are able to capture straight segments of the interface exactly and the error is accumulated only around the vertices. With  $O(1)$  mixed cells contributing to the average error,  $\Delta\omega$  is only  $O(h^2)$  (see (3)), which results in  $\tilde{\epsilon}_{avg} = \Delta\omega / (8\tilde{L}^2) = O(h^2)$  average deviation. Once again MoF algorithm demonstrates the highest accuracy. Since there is only one mixed cell per vertex of polygonal interface that contributes to MoF average error against three to four for competitors, MoF average error is at least two times lower than of any other method. We would also like to emphasize the superiority of MoF algorithm over the VoF-PLIC methods on a very coarse scale.

In terms of performance the clear winner is LSGQ, which is about two times faster than MoF and Swartz methods and six times faster than LVIRA. Common software components shared between the implementations of different methods allow us to claim a non-discriminative character of these benchmarks.

## 5 Dynamic tests

In presented dynamic tests interface is moving with prescribed solenoidal velocity field. Dynamic tests require development of algorithms for advancing in time of volume fractions for VoF-PLIC methods, as well as centroids in case of MoF method. These algorithms are presented in the beginning of Sections 5.1, 5.2.

Next we present numerical results for two test problems using our new MoF method as well as other LSG, LVIRA and Swartz methods.

In all tests we assume that velocity field  $\mathbf{v}(\mathbf{x}, t)$  is *solenoidal* and given analytically. The algorithm for advancing in time imposes no constraint on the Courant number  $CFL = v \Delta t/h$  ( $\Delta t$  is the time step,  $h$  is the local mesh spacing, and  $v$  is the local flow speed).

### 5.1 Time Advance of Volume Fractions

Given the location of mixed cell interfaces at  $t_{k-1} = \Delta t (k - 1)$ , one can evaluate the content of cells at  $t_k = t_{k-1} + \Delta t$  as follows:

```

for each cell  $\Omega_i$ ,  $i = \overline{1, N}$  do
    track  $\Omega_i$  back in time to  $t = t_{k-1}$  to identify the Lagrangian prototype  $\Omega_{i,k-1}$ 
                                                    of the cell;
    find the volume  $\tilde{m}_{i,k-1}$  of the reference phase enclosed in Lagrangian
                                                    prototype  $\Omega_{i,k-1}$ ;
    put the volume  $m_{i,k}$  of the reference component in  $\Omega_i$  equal to  $\tilde{m}_{i,k-1}$ .
end do

```

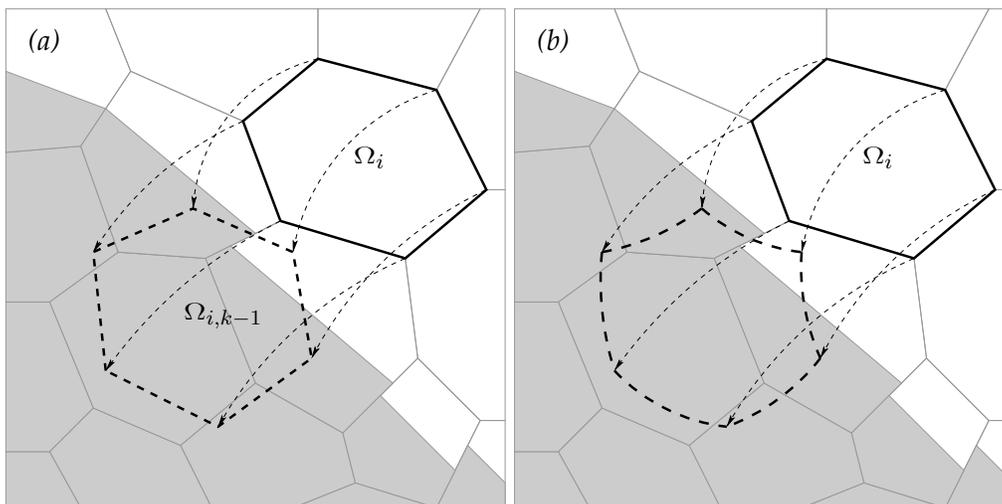
The vertices of polygonal cell  $\Omega_i$  are tracked back along the streamlines by means of the 4-th order Runge-Kutta scheme and then connected in proper order by the straight segments. This results in a polygon that we consider to be a *discrete Lagrangian prototype*  $\Omega_{i,k-1}$  of cell  $\Omega_i$  (see Figure 11).

Using a polygon intersection routine<sup>3)</sup>, we find the intersections of  $\Omega_{i,k-1}$  with all covered cell fractions at  $t = t_{k-1}$ . By partitioning the bounding box

---

<sup>3)</sup> Due to COnservative REMapper (CORE) library by M. Staley [17].

of the entire computational domain into 2D array of rectangular bins and pre-sorting all the mesh cells among these bins based on their centroid location, one can significantly accelerate the search of the cell fractions covered by the prototype. Whenever  $CFL \leq 1$ , it is sufficient to intersect the prototype only with  $\Omega_i$  and its direct neighbors. Moreover, if  $CFL \leq 1$  and  $\Omega_i$  along with all its neighbors is empty at  $t = t_{k-1}$ , then it is guaranteed to stay empty the next discrete moment of time, and there is no need to perform any polygon intersections in this case at all.



**Figure 11.** Discrete (a) and true (b) Lagrangian prototypes of the cell.

In *linear* velocity field, which is known to preserve straight lines, the *true Lagrangian prototype* of a polygonal cell is always a polygon, and the accuracy of the remapping is limited only by the accuracy the integration scheme. For a  $p$ -th order accurate scheme the *area defect*  $|\Omega_{i,k-1}| - |\Omega_i|$  is estimated as  $O(h \Delta t^{p+1})$ .

In a nonlinear velocity field straight lines are not preserved, and the Lagrangian prototype of a polygonal cell is not exactly a polygon. Therefore by ignoring the curvature of the prototype edges we introduce additional  $O(h^3 \Delta t)$  area defect.

Even a small area defect can eventually cause the algorithm to halt. Indeed, if the volume of the reference component  $\tilde{m}_{i,k-1}$  enclosed in the prototype  $\Omega_{i,k-1}$  exceeds the capacity of the cell  $|\Omega_i|$ , we are in trouble. Another, less critical, situation occurs when the prototype, being filled with the reference component, happens to have the area  $|\Omega_{i,k-1}|$  smaller than  $|\Omega_i|$ . In this case the cell becomes mixed, even though its prototype contains only one component.

In order to fix these flaws we use a post-remapping *repair* procedure. For every cell  $\Omega_i$ ,  $i = \overline{1, N}$  we specify the lower  $\underline{m}_{i,k}$  and the upper  $\overline{m}_{i,k}$  bounds of

the reference fraction volume  $m_{i,k}$  allowed:

$$\begin{aligned} \underline{m}_{i,k} &= \overline{m}_{i,k} = 0 && \text{if the prototype is empty,} \\ \underline{m}_{i,k} &= \overline{m}_{i,k} = |\Omega_i| && \text{if the prototype is full,} \\ \underline{m}_{i,k} &= 0, \overline{m}_{i,k} = |\Omega_i| && \text{otherwise,} \end{aligned} \quad (15)$$

and then force each volume  $m_{i,k}$ ,  $i = \overline{1, N}$  to fit into these bounds:

```

for each cell  $\Omega_i$ ,  $i = \overline{1, N}$  do
  if  $\Omega_i$  is overfilled ( $\overline{m}_{i,k} < m_{i,k}$ ) then
    try to redistribute the excess between the non-overfilled neighbours
    while there is still some excess remained do
      redistribute them among the next layer of the surrounding cells
    end do
  else if  $\Omega_i$  is underfilled ( $m_{i,k} < \underline{m}_{i,k}$ ) then
    try to compensate the shortfall by borrowing from the non-underfilled
    neighbours
    while there is still some shortage of material do
      borrow it from the next layer of the surrounding cells
    end do
  end if
end do

```

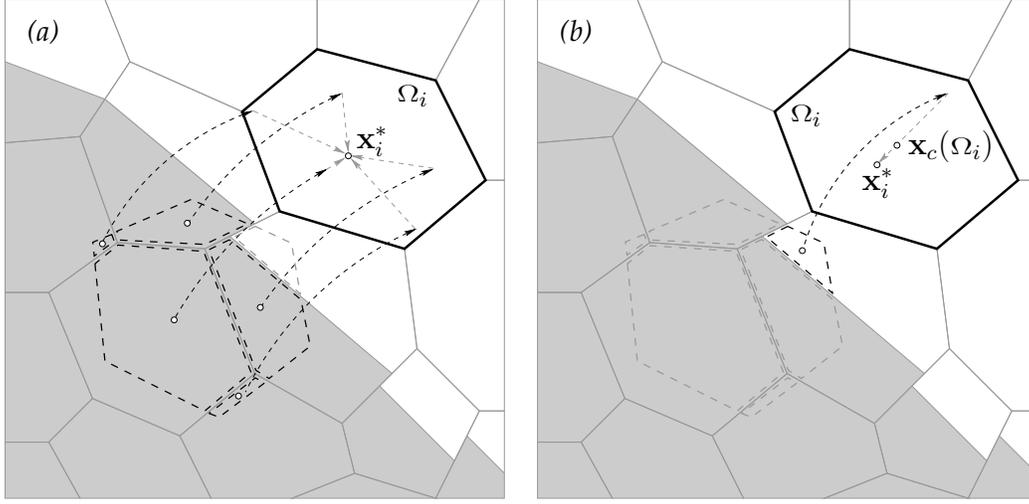
Due to the local nature of the area defect, the redistribution usually involves only direct neighbours of the cell. Therefore the complexity of the whole repair step comes to the total of  $O(N)$ .

## 5.2 Time Advance of Centroids

The technique above is easily extensible advancing centroid positions in time. Whenever a non-empty intersection of discrete Lagrangian prototype  $\Omega_{i,k-1}$  with any cell fraction is detected, one has to calculate not only the area but also the first moment of the intersection. After all the intersections are found, the centroid of the reference fraction of  $\Omega_{i,k-1}$  is calculated as the moment to volume ratio and is tracked forth along the streamlines to determine the position of the reference centroid in cell  $\Omega_i$ . There is no need to track the centroid if the prototype contains only one component: with (15) bounds the repair step guarantees that the cell will be pure as well.

Whenever the velocity field is linear in space, actual centroid velocity of any volume of fluid coincide with the field velocity at the centroid location:

$$\frac{d}{dt} \mathbf{x}_c(\omega) = \mathbf{v}(\mathbf{x}_c(\omega)). \quad (16)$$



**Figure 12.** Two way to improve the reference centroid accuracy: (a) by advecting the reference fraction in pieces; (b) by advecting the smaller complementary fraction.

Therefore, in this case, the trajectory of real centroid follows a streamline and the reference centroid error is due only to approximate integration ( $O(\Delta t^{p+1})$  for a  $p$ -th order scheme).

For a nonlinear velocity field the identity (16) is void. As long as  $\mathbf{v}(\mathbf{x}, t)$  is twice differentiable in  $\mathbf{x}$ , the following estimate, given by Taylor expansion, holds:

$$\frac{d}{dt}\mathbf{x}_c(\omega) = \mathbf{v}(\mathbf{x}_c(\omega)) + O(d^2). \quad (17)$$

Here  $d = \text{diam}(\omega)$  is the diameter of the volume advected. The cumulative reference centroid error in this case is  $O(d^2 \Delta t)$ .

Since the reference centroid error is determined by the diameter of the advected volume, some accuracy improvement can be gained by advecting the fluid in smaller pieces. The reference fraction of Lagrangian prototype is originally assembled from elementary intersections of  $\Omega_{i,k-1}$  with the reference fractions of cells on the previous time step. Therefore, instead of advecting the aggregated centroid, one may choose to advect centroids of these elementary parts separately and only then proceed with their aggregation (Figure 12a).

Also, whenever the volume of the complementary phase in the prototype is smaller than the volume of the reference component ( $|\Omega_{i,k-1} \setminus \bar{\omega}_{i,k-1}| < |\omega_{i,k-1}|$ ), one should advect the complementary centroid instead and then use the identity

$$\mathbf{x}_c(\omega_i) |\omega_i| + \mathbf{x}_c(\Omega_i \setminus \bar{\omega}_i) |\Omega_i \setminus \bar{\omega}_i| = \mathbf{x}_c(\Omega_i) |\Omega_i|$$

to get the position of the reference centroid (Figure 12b).

### 5.3 Numerical Results

**Example 1. - Solid Rotation of Letter A** The first set of snapshots (Figure 13) is due to the advection of the letter “A” glyph in the stationary rotation field:

$$\mathbf{v}((x, y), t) = \begin{bmatrix} -(y - y_0) \\ +(x - x_0) \end{bmatrix}, \quad (x_0, y_0) = (0.5, 0.5).$$

Other parameters of the numerical experiment:

$$\begin{aligned} \text{computational domain} & \quad ]0, 1[ \times ]0, 1[, \\ \text{simulation time} & \quad T = 2\pi, \\ \text{computational grid} & \quad 1024 \text{ polygonal cells,} \\ \text{number of time steps} & \quad N_T = 144. \end{aligned}$$

The velocity field is linear so that the error introduced by interface reconstruction prevails over the advection error. All VoF-PLIC methods oversmooth the interface, while MoF method demonstrates a good ability to preserve the shape of the glyph.

**Example 2 - Reversible vortex** The second test case [16] is about modeling evolution of the round “blot”

$$\{ \mathbf{x} \in \mathbb{R}^2 \mid \| \mathbf{x} - \mathbf{x}_0 \|_2 \leq R \}, \quad \mathbf{x}_0 = (0.5, 0.75), \quad R = 0.15$$

in the vortex field:

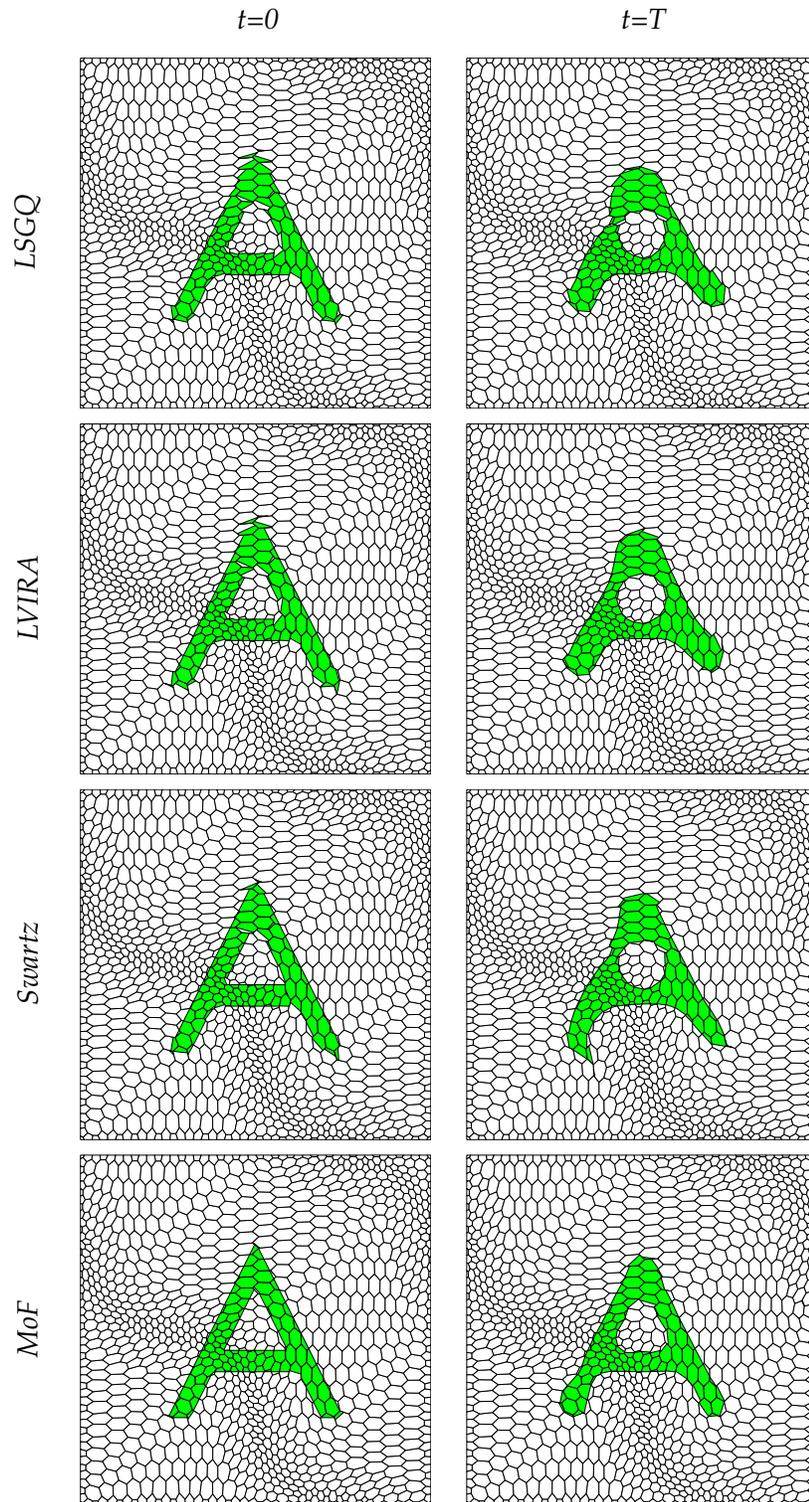
$$\mathbf{v}((x, y), t) = \begin{bmatrix} +\sin^2(\pi x) \sin(2\pi y) \\ -\sin^2(\pi y) \sin(2\pi x) \end{bmatrix} \cos(\pi t/T).$$

The cosine multiplier gradually “decreases the power” of the vortex until the complete stop ( $\mathbf{v}(\mathbf{x}, t) \equiv \mathbf{0}$ ) at  $t = T/2$ , and then starts to “power up” the reversed vortex. Since

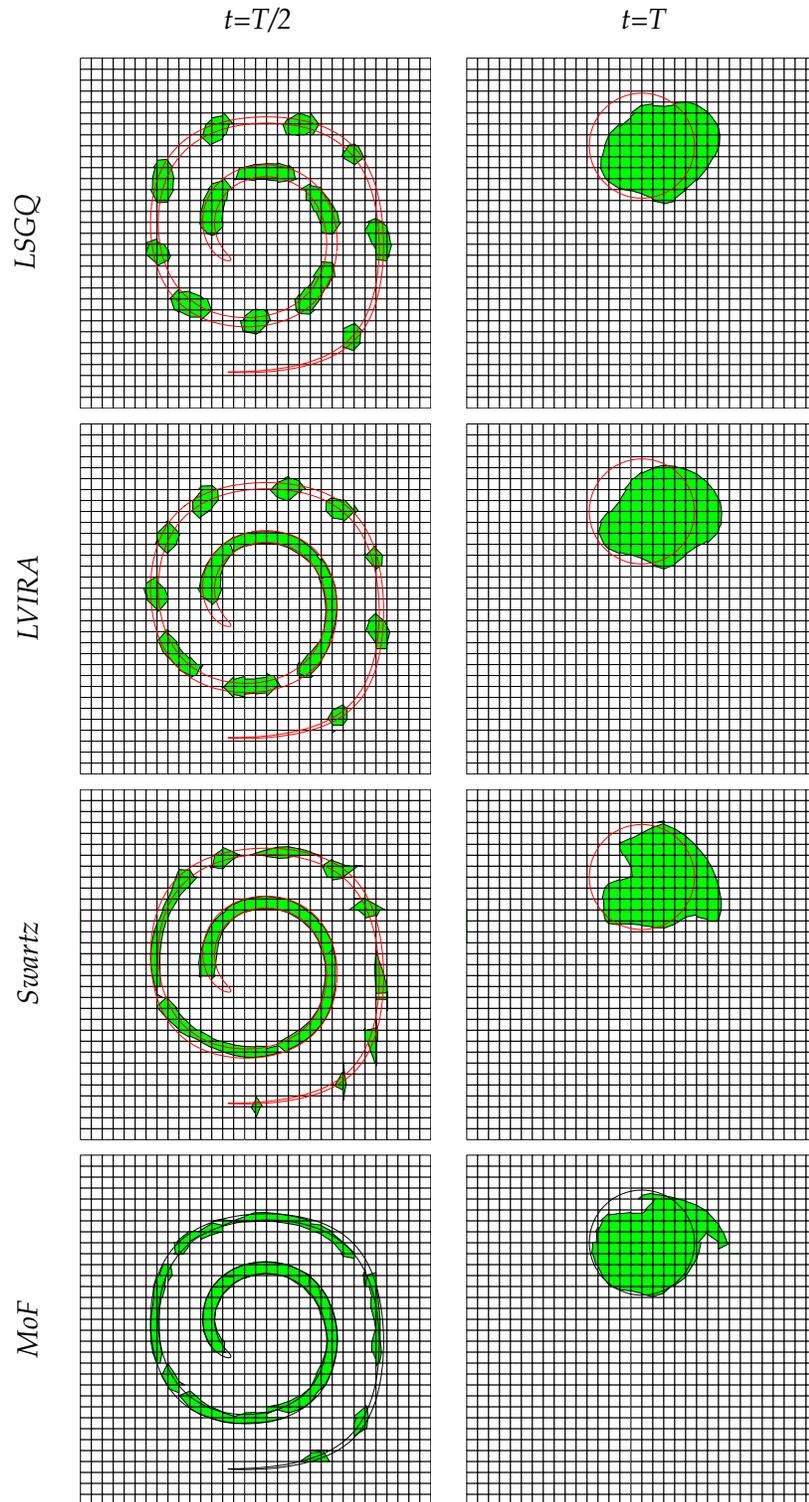
$$\mathbf{v}(\mathbf{x}, t) = -\mathbf{v}(\mathbf{x}, T - t), \quad 0 < t < T,$$

the exact integration of the fluid motion from  $t = 0$  to  $t = T$  should result in zero changes. Therefore by comparing the final configuration against the initial setup, one can get the idea of the accuracy of numerical method. For easier judgment we overlaid the snapshots (Figure 14) with the interface line obtained with tracking technique [1].

Other parameters of the experiment:



**Figure 13.** Rotation test results.



**Figure 14.** Reversible vortex test results.

computational domain	$]0, 1[ \times ]0, 1[$ ,
simulation time	$T = 8$ ,
computational grid	uniform, $32 \times 32$ cells,
number of time steps	$N_T = 256$ .

Although the final shape obtained with MoF method demonstrates better overlap with the initial shape than the VoF-PLIC alternatives, the latter result in much smoother interface. The reason for this is that VoF methods, due to the aggregate nature of the interface normal estimate, have a tendency to oversmooth the interface, which on the dynamic level can be interpreted as an artificial numerical surface tension. This feature becomes determinative when a VoF method is forced to work on the edge of the resolution limit: by the moment of maximum stretch ( $t = T/2$ ) the shape of the reference component becomes so thin that it breaks into a series of separate blots. MoF method does not exhibit this kind of behavior and gives much better approximation of real interface at  $t = T/2$ .

## 6 Acknowledgment

Authors would like to thank Blair Swartz and Rao Garimella from Mathematical Modeling and Analysis Group (T-7) at LANL for their sincere interest in the subject and valuable remarks that guided this research in the right direction. Authors also thank M. Staley, B. Rider, R. Lowrie, D. Kothe, M. Francois, J. Mosso, T. Aslam and D. Benson for useful discussions.

The authors also acknowledge the support of ASC program at Los Alamos National Laboratory.

This work was performed at the Los Alamos National Laboratory operated by the University of California for the US Department of Energy under the contract W-7405-ENG-36. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of publication or guarantee its technical correctness.

## 7 Appendix

All VoF-PLIC algorithms introduced above rely on the list of adjacent cells that involved in evaluation of the interface normal in a mixed cell. We consider two cells to be adjacent if they share an edge or a vertex.

Implementations of LSGQ and LVIRA follow exactly the descriptions given in [2] and [14] respectively. The minimum of LVIRA objective function is considered to be localized when the polar angle of the outward normal iterate stabilizes within  $tol_\phi = 10^{-6}$  radians.

Our implementation of Swartz algorithm is somewhat different from the original description given in [10]. This is actually the reason why we call it *Swartz*, not *Mosso-Swartz*.

The idea of the method is based on the fact that for a pair of adjacent mixed cell there exists a common linear interface that satisfies given volume fractions<sup>4)</sup>. This common linear interface can be found by means of the following iterative procedure. Given initial VoF-PLIC approximation in both mixed cells, one has to:

- 1) connect the medians of interfaces with a straight segment;
- 2) use the normal of this segment as a new iterate for the normal of common linear interface;
- 3) build a new linear interface in each cell using this normal;
- 4) repeat this cycle until the direction of the common interface normal establishes within some small tolerance (we used  $tol_\phi = 10^{-6}$ ).

The direction on the interface normal in a given mixed cell is defined as an average of common interface normals due to all possible mixed neighbours.

In addition to this procedure, Mosso-Swartz algorithm [10] specifies an *external iterative process* that runs through the list of mixed cell and repeats the procedure above in *Gauss-Seidel manner*, updating the interface normal in a mixed cell with the new averaged one as soon as the latter is evaluated. We came to conclusion that such an external iterations result in excessive numerical surface tension, oversmooth the interface and decrease the resolution of the algorithm. Therefore in our implementation we do not use external iterations and also update the interface normals in *Jacobi manner*, i.e. only after all the normals have been evaluated from the original initial guess.

There are several important details about *internal iterations* that were not discussed in the original paper.

---

<sup>4)</sup> For essential prerequisites see [21].

- A *common interface* for a given pair of adjacent cells and respective volume fractions *is not unique*. The iterative process may converge to either of them, depending on initial guess and implementation strategy.
- There are *two choices of the outward normal* on the segment that connects the medians of adjacent interfaces. What would be a reasonable deterministic rule for the choice?
- Are the common interfaces due to different neighbours averaged with equal weights or not?

It is clear that different choices of initial guess and implementation strategy result in different PLIC reconstructions for the same set of volume fractions. We resolve the issues above as follows:

- Our implementation of Swartz algorithm is seeded with the interfaces given by LSGQ method.
- We do not proceed with internal iterations for a pair of adjacent mixed cells if the initial outward normals differ by more than 45 degrees. Between the two opposite normals of the segment connecting the interface medians we choose the one that has a positive projection on both initial outward normals. If neither of two possible normals satisfy this criterion, the iterations are aborted with void result.

The rationale of this strategy comes clear if we start to look at Swartz iterations as at a second order accurate “upgrade” of initial interface approximation: if two initial normals already differ significantly, then the common linear interface is highly unlikely to give an adequate approximation to the original one, and therefore should not contribute to the average. This way the “upgrade” kicks in only for those pairs of adjacent cells that carry information about the same smooth segment of original interface.

- All common normals are averaged with equal weights.

One may argue that our interpretation of Swartz algorithm is subjective and the quality of interface reconstruction it provides is not superior to the original Mosso-Swartz version. We may assure the reader that we tested a lot of variations of the algorithm and found the one described above to be the most robust.

## References

- [1] E. Aulisa, S. Manservigi, and R. Scardovelli. A surface marker algorithm coupled to an area-preserving marker redistribution method for three-dimensional interface tracking. *Journal of Computational Physics*, 197(2):555–584, Jul 2004.
- [2] T. J. Barth and D. C. Jespersen. The design and application of upwind schemes on unstructured meshes. In *Proceedings of the 27th Aeroscience Meeting*, Reno, NV, Jan 1989. AIAA.
- [3] David J. Benson. Eulerian finite element methods for micromechanics of heterogeneous materials: Dynamic prioritization of material interfaces. *Computer Methods in Applied Mechanics and Engineering*, 151:343–360, 1998.
- [4] David J. Benson. Volume of fluid interface reconstruction methods for multi-material problems. *Appl. Mech. Rev.*, 55(2):151–165, March 2002.
- [5] J. Frédéric Bonnans, J. Charles Gilbert, Claude Lemaréchal, and Claudia A. Sagastizábal. *Numerical Optimization: Theoretical and Practical Aspects*. Springer-Verlag, Berlin Heidelberg, 2003.
- [6] R. DeBar. Fundamentals of the KRAKEN code. Technical Report UCID-17366, Lawrence Livermore National Laboratory, 1974.
- [7] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183:83–116, 2002.
- [8] W. H. McMaster. Computer codes for fluid-structure interactions. Technical Report UCRL-89724, Lawrence Livermore National Laboratory, 1984.
- [9] J. P. Morris. Simulating Surface Tension with Smoothed Particle Hydrodynamics. *International Journal for Numerical Methods in Fluids*, 33:333–353, 2000.
- [10] S. J. Mosso, B. K. Swartz, D. B. Kothe, and R. C. Ferrell. A Parallel, Volume-Tracking Algorithm for Unstructured Meshes. In P. Schiano, A. Ecer, J. Peiriaux, and N. Satofuka, editors, *Parallel Computational Fluid Dynamics: Algorithms and Results Using Advanced Computers*, pages 368–375. Elsevier Science B.V., 1997.
- [11] Stewart Mosso and Sean Clancy. A Geometrically Derived Priority System for Young’s Interface Reconstruction. Technical Report LA-CP-95-0081, Los Alamos National Laboratory, 1995.
- [12] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer-Verlag, New York, 1999.

- [13] S. Osher and R. P. Fedkiw. Level Set Methods: An Overview and Some Recent Results. *Journal of Computational Physics*, 169(2):463–502, May 2001.
- [14] J. E. Pilliod and E. G. Puckett. Second-order Accurate Volume-of-Fluid Algorithms for Tracking Material Interfaces. *Journal of Computational Physics*, 199(2):465–502, Sep 2004.
- [15] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian T. Flannery. *Numerical Recipes in C: the Art of Scientific Computing, Second Edition*. Cambridge University Press, New York, 1992.
- [16] W. J. Rider and D. B. Kothe. Reconstructing Volume Tracking. *Journal of Computational Physics*, 141:112–152, 1998.
- [17] Martin Staley. CORE: Conservative Remapper. Technical Report LA-UR-04-8104, Los Alamos National Laboratory, 2004.
- [18] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome. An Adaptive Level Set Approach for Incompressible Two-Phase Flows. *Journal of Computational Physics*, 148:81–124, 1999.
- [19] M. Sussman, E. Fatemi, P. Smereka, and S. Osher. Improved Level Set Method for Incompressible Two-Phase Flows. *Computers & Fluids*, 27(5-6):663–680, Jun-Jul 1998.
- [20] M. Sussman and E. G. Puckett. A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows. *Journal of Computational Physics*, 162(2):301–337, Aug 2000.
- [21] B. Swartz. The Second-Order Sharpening of Blurred Smooth Borders. *Mathematics of Computation*, 52(186):675–714, Apr 1989.
- [22] H. S. Udaykumar and W. Shyy. Grid-Supported Marker Particle Scheme for Interface Tracking. *Numerical Heat Transfer, Part B (Fundamentals)*, 27(2):127–153, Mar 1995.
- [23] D. L. Youngs. Time-Dependent Multi-Material Flow with Large Fluid Distortion. In K. W. Morton and M. J. Baines, editors, *Numerical Methods for Fluid Dynamics*, pages 273–285. Academic Press, 1982.
- [24] D. L. Youngs. An Interface Tracking Method for a 3D Eulerian Hydrodynamics Code. Technical Report 44/92/35, AWRE, 1984.