Journal of Computational Physics 328 (2017) 131–139

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Moment-of-fluid analytic reconstruction on 2D Cartesian grids

Antoine Lemoine[a,*], Stéphane Glockner[b], Jérôme Breil[c]

[a] Univ. Bordeaux, I2M, UMR 5295, F-33400 Talence, France
[b] Bordeaux INP, I2M, UMR 5295, F-33400 Talence, France
[c] CEA CESTA, 15 Avenue des Sablières, CS 60001, 33116 Le Barp Cedex, France

ARTICLE INFO

Article history:
Received 17 December 2015
Received in revised form 13 July 2016
Accepted 5 October 2016
Available online 12 October 2016

Keywords:
Moment-of-Fluid
Interface reconstruction
Cartesian grid

ABSTRACT

Moment-of-Fluid (MoF) is a piecewise linear interface reconstruction method that tracks fluid through its volume fraction and centroid, which are deduced from the zeroth and first moments. We present a method that replaces the original minimization stage by an analytic reconstruction algorithm on bi-dimensional Cartesian grids. This algorithm provides accurate results for a lower computational cost than the original minimization algorithm. When more than two fluids are involved, this algorithm can be used coupled with the minimization algorithm. Although this paper deals with Cartesian grids, everything remains valid for any meshes that are made of rectangular cells.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Moment-of-Fluid (MoF) [1–12] is a method to represent and reconstruct interfaces in multiphase flow simulation. This method is the apex of the Volume-of-Fluid (VoF) methods using piecewise linear interface reconstruction [13,14]. MoF represents the interfaces with the first two moments of any material subset ω in a polygonal cell Ω, namely the volume $M_0(\omega)$ and the first momentum $\boldsymbol{M}_1(\omega)$.

$$M_0(\omega) = \int_\omega d\boldsymbol{x} \qquad \boldsymbol{M}_1(\omega) = \int_\omega \boldsymbol{x} d\boldsymbol{x} \tag{1}$$

Sometimes it is more convenient to use their relative equivalent quantities, namely the volume fraction $\mu(\omega)$ and the centroid $\boldsymbol{x}_c(\omega)$.

$$\mu(\omega) = \frac{M_0(\omega)}{M_0(\Omega)} \qquad \boldsymbol{x}_c(\omega) = \frac{\boldsymbol{M}_1(\omega)}{M_0(\omega)} \tag{2}$$

MoF consists in finding a polygonal approximation ω^ℓ of a reference subset $\omega^\star$ (see Fig. 1). The part of the boundary $\Gamma^\ell = \partial\omega^\ell \setminus \partial\Omega$ is an affine approximation of the reference interface $\Gamma^\star = \partial\omega^\star \setminus \partial\Omega$. Furthermore, ω^ℓ verifies the following minimization problem:

$$\text{Find} \quad \omega^\ell = \underset{\omega^\ell}{\operatorname{argmin}} \, |\boldsymbol{x}_c(\omega^\ell) - \boldsymbol{x}_c(\omega^\star)|^2 \qquad \text{such that} \qquad M_0(\omega^\ell) = M_0(\omega^\star) \tag{3}$$

* Corresponding author.
E-mail addresses: antoine.lemoine@bordeaux-inp.fr (A. Lemoine), glockner@bordeaux-inp.fr (S. Glockner), jerome.breil@u-bordeaux.fr (J. Breil).

http://dx.doi.org/10.1016/j.jcp.2016.10.013

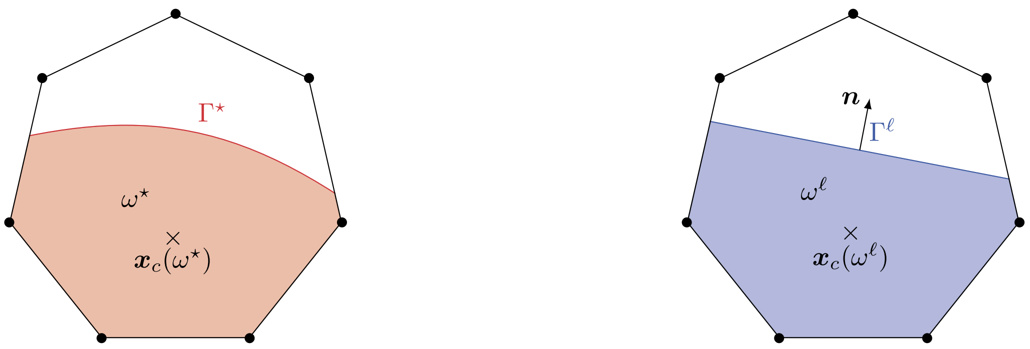

Fig. 1. Reference subset (left) and reconstructed subset (right).

In the remainder of this article, we use the following notations: the geometric elements with a $\star$ in exponent refer to the reference configuration, the geometric elements with an ℓ in exponent refer to the solution of problem (3) and the geometric elements without an exponent refer to any affine approximation.

Since the seminal publications [1–4], MoF has been adapted to specific applications. For instance, MoF has been exploited in an adaptive mesh refinement (AMR) context [5], used in arbitrary Lagrangian–Eulerian (ALE) schemes [6,8], coupled with level-set representation of multiphase flow [9], used in compressible multiphase flow [11], and adapted to axisymmetric coordinates [15] or in cylindrical geometry with an ALE approach [16]. Moreover, the accuracy of MoF has been improved thanks to the symmetric reconstruction [10] or with the introduction of filament capturing [12].

This article focuses on the implementation of MoF reconstruction on bi-dimensional Cartesian grids or any meshes composed of rectangular cells, which are widely used in computational fluid dynamics. While classic implementations use a time consuming minimization algorithm to solve the minimization problem, we propose a faster analytic reconstruction formula which takes advantage of the shape of the cells. The next section presents the demonstration of this analytic formula in two steps. First, we propose a parametrization of the locus of the centroids in a fixed volume. We show that this curve can be parametrized by four parabolas and four hyperbolas. Second, we find the minimal distance of the reference centroid to the curve with an orthogonal projection. This involves the computation of the minimal distance from a point to a hyperbola and to a parabola. In the remainder of this article, in section 3, we propose an analytical algorithm where we have reduced the search of the minimal distance to two parabolas and one hyperbola. In section 4, we discuss about existence and uniqueness of a solution. In section 5, we compare the proposed algorithm to the original minimization algorithm.

2. Analytic reconstruction

2.1. Description

Consider a rectangular cell of dimension (c_x, c_y) such as represented in the center of Fig. 2. The locus of the centroids $\boldsymbol{x}_c(\omega) := (g_x, g_y)$ for a given reference volume $V := M_0(\omega^\star)$ is a closed convex curve. In Fig. 2, we have represented the locus of the centroids for various reference volumes such that $V \le 0.5 c_x c_y$. We observe 8 different configurations. 4 configurations where the reconstructed polygon is a triangle (odd numbers on the figure) and 4 configurations where the reconstructed polygon is a quadrangle (even numbers on the figure). Only the first two configurations can be considered since any other configuration can be transformed into the first two by symmetry and/or inverting the role of c_x and c_y. In paragraph §2.2, we prove that when the reconstructed polygon is a triangle, the locus is a hyperbola $\mathcal{H}$ and when the reconstructed polygon is a quadrangle, the locus is a parabola $\mathcal{P}$. When $V > 0.5 c_x c_y$, the method can be applied on the dual (or complementary) configuration $\widetilde{\omega}^\star := \Omega \setminus \omega^\star$, that is, we consider the volume $M_0(\widetilde{\omega}^\star) = M_0(\Omega) - V$ and the first momentum $\boldsymbol{M}_1(\widetilde{\omega}^\star) = \boldsymbol{M}_1(\Omega) - \boldsymbol{M}_1(\omega^\star)$.

2.2. Parametrization

The reconstructed line segment Γ can be defined with two parameters, namely the interface normal (n_x, n_y) and the distance to the origin ξ, that is:

$$\Gamma = \{(x, y) \in \Omega \ / \ n_x x + n_y y = \xi\} \tag{4}$$

In the triangle configuration, Γ intersects the bottom edge of the cell in $0 < \alpha \le c_x$ and the left edge of the cell in $0 < \beta \le c_y$ (see Fig. 3). For a given reference volume V and a normal (n_x, n_y), the intersection coordinates are given by:

$$\alpha = \sqrt{2V \frac{n_y}{n_x}} \qquad \beta = \sqrt{2V \frac{n_x}{n_y}} \tag{5}$$

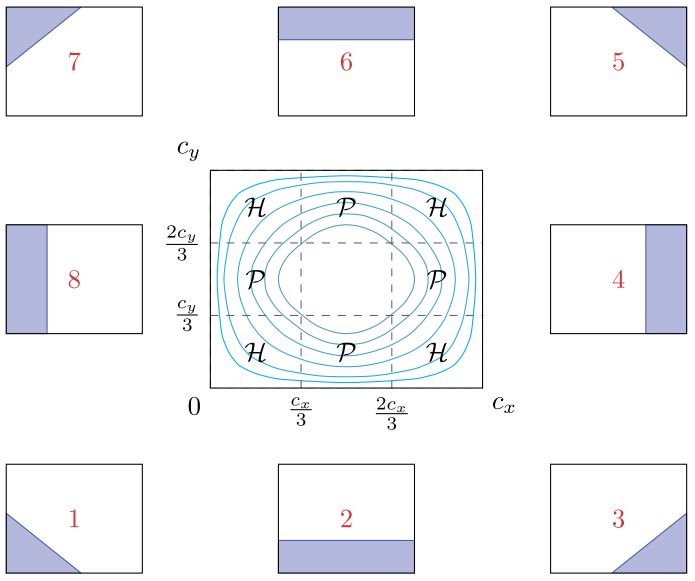

Fig. 2. Representation of the different configurations of the locus of the centroids (cyan) in a rectangular cell of dimension (c_x, c_y) for various reference volumes V such that $V \leq 0.5c_xc_y$. Depending on the reconstructed polygon, the locus is either a hyperbola $\mathcal{H}$ or a parabola $\mathcal{P}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

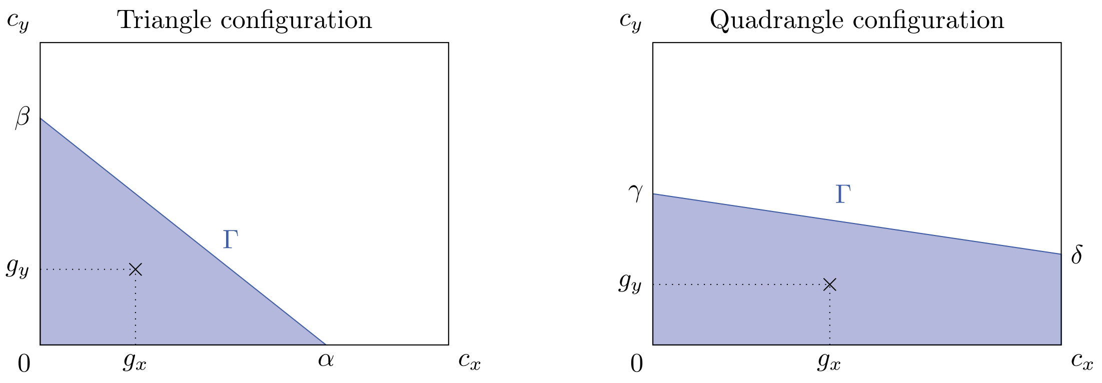

Fig. 3. Parametrization of triangle and quadrangle configurations.

The coordinates of the triangle centroid are given by:

$$g_x = \frac{1}{3}\sqrt{2V\frac{n_y}{n_x}} \qquad g_y = \frac{1}{3}\sqrt{2V\frac{n_x}{n_y}} \tag{6}$$

To express g_y as a function of g_x, we substitute n_x/n_y which gives an equation of a hyperbola:

$$g_y := \mathcal{H}(g_x) = \frac{2V}{9}\frac{1}{g_x} \tag{7}$$

The validity domain of this formula is given for the limit cases when $\beta = c_y$ and when $\alpha = c_x$. Simple calculations give:

$$g_x \in \left[\frac{2V}{3c_x}, \frac{c_x}{3}\right] \tag{8}$$

In the quadrangle configuration, Γ intersects the left edge of the cell in $0 < \gamma \leq c_y$ and the right edge of the cell in $0 < \delta \leq c_y$ (see Fig. 3). For a given reference volume V and a normal (n_x, n_y), the intersection coordinates are given by:

$$\gamma = \frac{V}{c_x} + \frac{n_xc_x}{2n_y} \qquad \delta = \frac{V}{c_x} - \frac{n_xc_x}{2n_y} \tag{9}$$

The coordinates of the quadrangle centroid are given by:

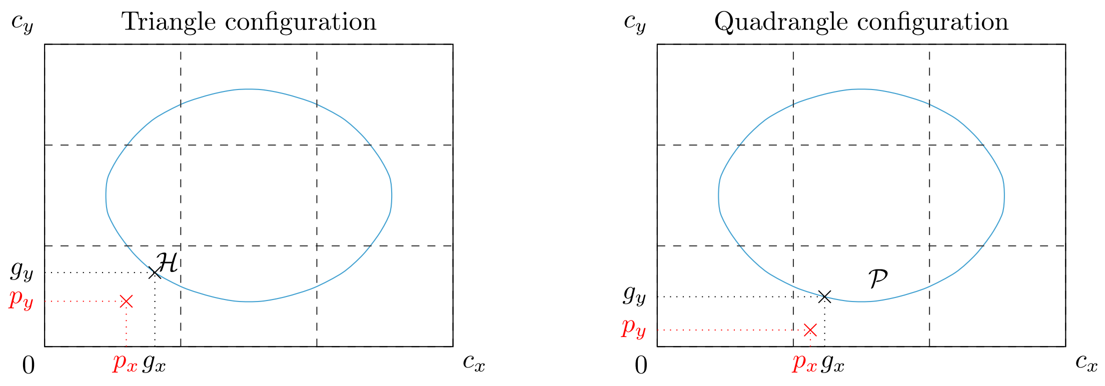

Fig. 4. Find the closest point (g_x, g_y) of a point (p_x, p_y) to the locus of the centroid in triangle and quadrangle configurations.

$$g_x = \left(\frac{1}{2} - \frac{c_x^2}{12V}\frac{n_x}{n_y}\right)c_x \qquad g_y = \left(\frac{V}{2c_x^2} + \frac{c_x^2}{24V}\frac{n_x^2}{n_y^2}\right)c_x \tag{10}$$

To express g_y as a function of g_x, we substitute n_x/n_y which gives an equation of a parabola:

$$g_y := \mathcal{P}(g_x) = \frac{V}{2c_x} + \frac{6V}{c_x}\left(\frac{1}{2} - \frac{g_x}{c_x}\right)^2 \tag{11}$$

The validity domain of this formula is given when the curve is not a hyperbola, *i.e.*:

$$g_x \in \left[\frac{c_x}{3}, \frac{2c_x}{3}\right] \tag{12}$$

Note that it is easy to check that the global curve defined by the union of all the parabolas and hyperbolas is three times differentiable, but the third derivative is not continuous.

We have shown that the locus of the centroid can be parametrized by four parabolas and four hyperbolas. In the next paragraph, we present how to find the minimal distance from the reference centroid to the various parts of the curve to find the global minimum.

2.3. Minimal distance

The minimal distance from any point $(p_x, p_y) \in \mathbb{R}^2$ to the locus of the centroid is its closest orthogonal projection (see Fig. 4). Since the curve is defined by parts, we need to express the minimal distance from any point to each part. The orthogonal projection $(g_x, g_y) = (g_x, \mathcal{H}(g_x))$ of the point (p_x, p_y) on the hyperbola $\mathcal{H}$ verifies the following equation.

$$(g_x - p_x, \mathcal{H}(g_x) - p_y) \cdot (1, \mathcal{H}'(g_x)) = 0 \tag{13}$$

This equation is the expression of the orthogonality of the vector defined by the point and its orthogonal projection with the tangent to the curve. Replacing $\mathcal{H}$ by its value (7) reveals a quartic equation on the coordinate g_x such that one of the real roots is the closest orthogonal projection.

$$g_x^4 - p_x g_x^3 + \frac{2}{9}V p_y g_x - \left(\frac{2V}{9}\right)^2 = 0 \tag{14}$$

The same process can be applied to find the orthogonal projection $(g_x, g_y) = (g_x, \mathcal{P}(g_x))$ of the point (p_x, p_y) on the parabola $\mathcal{P}$. The orthogonal projection verifies the following equation.

$$(g_x - p_x, \mathcal{P}(g_x) - p_y) \cdot (1, \mathcal{P}'(g_x)) = 0 \tag{15}$$

Substituting $\mathcal{P}$ by its expression (11) unveils a cubic equation on g_x.

$$g_x - p_x - \frac{12V}{c_x^2}\left(\frac{1}{2} - \frac{g_x}{c_x}\right)\left(\frac{V}{2c_x} - p_y\right) - \frac{72V^2}{c_x^3}\left(\frac{1}{2} - \frac{g_x}{c_x}\right)^3 = 0 \tag{16}$$

The dependence of equations (14) and (16) on so many parameters (p_x, p_y, c_x, c_y and V) makes it tricky to find the root that gives the closest orthogonal projection. The simplest way to find it is to compute all the roots and eliminate the roots outside their defined domain and find the one that gives the closest orthogonal projection. In our implementation, we use the algorithms proposed in [17] and [18] to accurately compute the cubic and quartic roots.

In the following paragraph, we define the transformations to solve the problem on any configuration.

2.4. Generalization to any configuration

As mentioned in paragraph §2.1, we have only considered a "reference configuration" where $V \leq 0.5 c_x c_y$ and where the reference centroid belongs to one of the two first configurations (1 and 2) represented on Fig. 2. Here, we provide the formulas to transform any configuration into one of these two reference configurations and to transform the results back to the original configuration. In the following, we denote by $\boldsymbol{x}_c(\omega^\star) = (p_x, p_y)$ the reference centroid. All the coordinates are expressed in the cell referential where the origin corresponds to the bottom left corner.

We consider that we have two algorithms to handle problems 1 and 2 that take the cell dimensions (c_x, c_y) and the coordinates of the reference centroid (p_x, p_y) as input parameters. The output data are the coordinates of the normal (n_x, n_y). Using some transformations, it is possible to use this algorithm to solve any other problems.

If the reference volume is greater than the half of the cell volume $V > 0.5 c_x c_y$, consider the dual volume $M_0(\widetilde{\omega}^\star) = c_x c_y - V$ and the dual reference centroid $\boldsymbol{x_c}(\widetilde{\omega}^\star) = (c_x c_y (c_x/2, c_y/2) - V(p_x, p_y))/(c_x c_y - V)$.

For problems 3 to 8, use the following transformations for input parameters and output data:

- Problems 3 and 4: cell dimensions (c_y, c_x), centroid coordinates $(p_y, c_x - p_x)$, normal $(-n_y, n_x)$.
- Problems 5 and 6: cell dimensions (c_x, c_y), centroid coordinates $(c_x - p_x, c_y - p_y)$, normal $(-n_x, -n_y)$.
- Problems 7 and 8: cell dimensions (c_y, c_x), centroid coordinates $(c_y - p_y, c_x)$, normal $(n_y, -n_x)$.

In the next section, we use the above preliminary work to propose an algorithm that replaces the original minimization algorithm to find the minimal distance from the reference centroid to the locus of the centroids.

3. Algorithm

The following algorithm computes the closest point $\boldsymbol{x}_c(\omega^\ell) = (g_x, g_y)$ of any reference centroid $\boldsymbol{x}_c(\omega^\star) = (p_x, p_y)$ on the locus of the centroids for a given volume V.

Step 1. If the volume of fluid is greater than the half of the cell volume $V > 0.5 c_x c_y$, consider the dual volume $M_0(\widetilde{\omega}^\star) = c_x c_y - V$ and the dual reference centroid $\boldsymbol{x}_c(\widetilde{\omega}^\star)$.
Step 2. Find the subset of the plane where the reference centroid is located. According to the results listed below, consider the corresponding problems listed on Fig. 2:
- if $p_x \leq 0.5 c_x$ and $p_y \leq 0.5 c_y$ consider the problems 1, 2 and 8;
- if $p_x > 0.5 c_x$ and $p_y \leq 0.5 c_y$ consider the problems 2, 3 and 4;
- if $p_x > 0.5 c_x$ and $p_y > 0.5 c_y$ consider the problems 4, 5 and 6;
- if $p_x \leq 0.5 c_x$ and $p_y > 0.5 c_y$ consider the problems 6, 7 and 8.

Step 3. Solve one quartic (14) and two cubic (16) equations of the considered problems.
Step 4. Eliminate every root that gives a result which is outside the domain defined in equations (8) and (12).
Step 5. Find the root that gives the closest orthogonal projection.
Step 6. If the dual problem was solved, transform the result back to the primal problem.

Note that the third step requires performing the transformations defined in §2.4 in order to apply the formula (14) and (16).

After finding the coordinates of the centroid $\boldsymbol{x}_c(\omega^\ell)$, formula (6) and (10) can be used to obtain the normal (n_x, n_y) or any other quantity to represent the reconstructed interface.

4. Discussion on existence and uniqueness

In article [1], the authors address a particular attention to uniqueness and existence of a solution. The method proposed in the present article solves the same minimization problem (3) as defined in [1]. As a consequence, the results about uniqueness and existence of [1] are also valid for the proposed method, that is, the solution of MoF interface reconstruction exists and is unique except for a set of reference centroids that have zero area. Thus, a particular attention must be paid to how multiple solutions are handled and how the solutions are eliminated in the fourth step of the proposed algorithm.

To the question about how to handle multiple solutions, the answer is implementation dependent. The proposed algorithm consists in computing all the valid solution for every selected configurations and selects the solution that gives the closest distance to the reference centroid. In our implementation, if there are two solutions that give the same closest distance, only the first solution is kept. As regards the minimization, the choice between multiple solution depends on the initial condition of the minimization algorithm. Therefore, in both algorithms, the choice is in some ways arbitrary.

To the question about how to eliminate solutions that are outside their definition domain, a particular attention must be paid when the solutions are close to the border of their definition domain. It is possible due to the finite precision and numerical errors that a correct solution is eliminated because it lies out, but very close, to the border of its definition domain. A possible answer to this problem consists in giving a thickness to the border. That corresponds to accept the solutions that are outside their definition domain, but close to the border with a given epsilon.

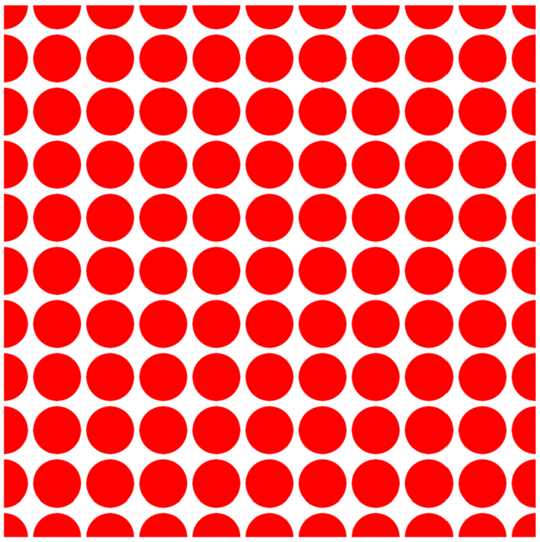

Fig. 5. MoF reconstruction of the static test case on a 2048^2 Cartesian grid.

Table 1
Time ratio minimization/analytic for the static test case on a 2048^2 Cartesian grid.

	Min. 10^{-15}	Min. 10^{-12}	Min. 10^{-9}	Min. 10^{-6}
Ana.	2.59	2.23	1.87	1.44

5. Test cases

We have implemented the proposed algorithm in the open-source code Notus (http://notus-cfd.org). We have tested this implementation on various cases from exact reconstruction to advected cases. We have compared the computational time with the standard minimization algorithm [1]. We consistently observe better performances for the analytic reconstruction, which span from 20% to 300% faster than the minimization method. This large range of performance results from the dependency of the method on a lot of parameters such as the tolerance value, the shape of the interface, the number of cells, and the number of fluids.

We have selected two test cases to provide a sample of the performance of this method. The first case is a static reconstruction on a shape that can be exactly reconstructed, *i.e.* the reference centroid belongs to the locus of the centroids. The second case is dynamic. It is closer to a practical case and will provide a better hint of the true performance than the static case.

In our implementation of the minimization algorithm, we use the line search algorithm [19] as recommended by [2]. The speed of the minimization algorithm depends on the Flood Algorithm used to reconstruct the interface from the normal and the volume fraction. In our implementation, we use the Flood Algorithm presented in [20] which is similar to the one presented in [1] but behaves better when two sections are very close to each other. The reader can refer to Appendix A for implementation details.

Remark. The minimization algorithm stops when the error reaches a prescribed tolerance value. Since the algorithm consists in bracketing the angle, the error is defined as the difference between the upper and lower bound of the bracketing. Note that this error should not be confused with the centroid defect. In other words, even if the centroid defect can not be zero, the bracketing error can always be as close as possible to zero, up to the machine error.

5.1. Static reconstruction

The static case consists in the reconstruction of many disks of radius 0.045 in a unit square domain $[0, 1]^2$ such as represented on Fig. 5. The disks are discretized such that an exact reconstruction is possible; the circle is approximated by straight line segments in the cells intersected by the interface. We use a Cartesian grid composed of 2048^2 cells to ensure a substantial computational time.

Table 1 presents the time ratio between minimization and analytic reconstruction for tolerance values 10^{-15}, 10^{-12}, 10^{-9} and 10^{-6}. These tolerance values are based on the angle, as defined in the previous remark.

In our experiments, we have observed that the centroid defect $|\boldsymbol{M}_1(\omega^\ell) - \boldsymbol{M}_1(\omega^\star)|$ given by the analytic reconstruction is on the order of the machine precision (10^{-16}). If we compare the computational time of the minimization process with

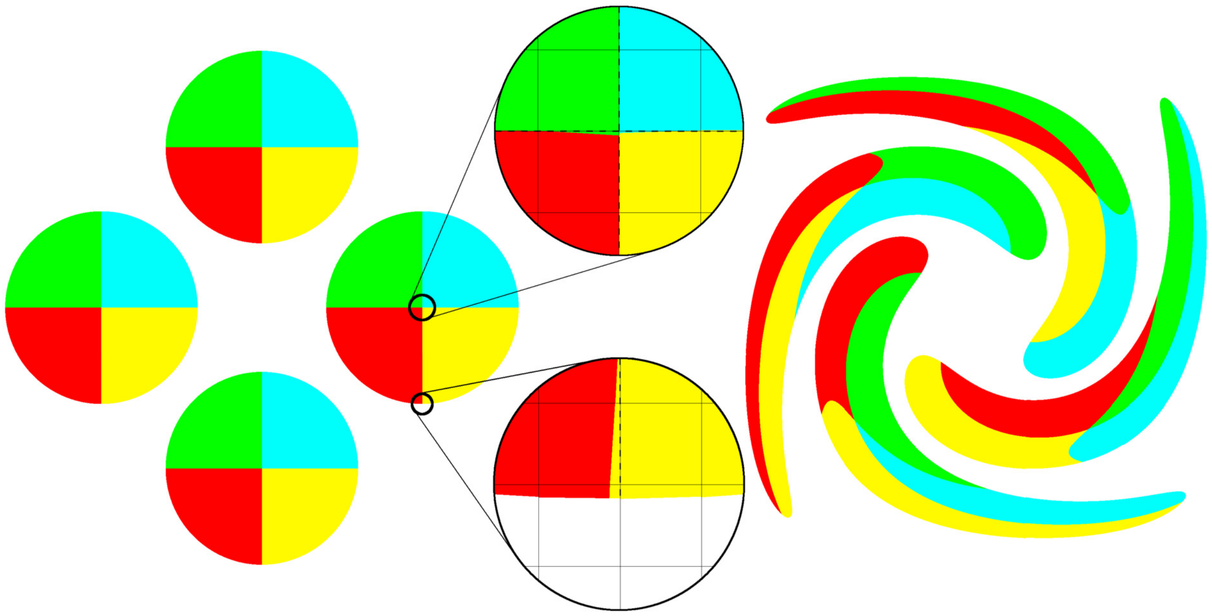

Fig. 6. MoF reconstruction of the dynamic test case on a 128^2 Cartesian grid. The left picture represents the initial and the final state and the right picture represents the maximal deformation. Magnified zones emphasize the defects at the final state compared to the initial state represented by dashed lines.

Table 2
Time ratio minimization/{analytic & minimization} for the dynamic test case on a 128^2 Cartesian grid.

	Min. 10^{-15}	Min. 10^{-12}	Min. 10^{-9}	Min. 10^{-6}
Ana. & Min. 10^{-15}	2.27	1.95	1.59	1.33
Ana. & Min. 10^{-12}	2.38	2.04	1.67	1.39
Ana. & Min. 10^{-9}	2.52	2.16	1.77	1.47
Ana. & Min. 10^{-6}	2.67	2.29	1.87	1.56

a tolerance value that gives an equivalent precision for the centroid defect, we observe that the analytic reconstruction is about 160% faster. With a tolerance value that would be used for a physical simulation (about 10^{-6}), we observe that the analytic reconstruction is at least 44% faster, but the centroid defect does not reach the machine error.

5.2. Dynamic reconstruction

In this numerical test we consider four disks, all composed of four materials, immersed in a reversible sheared flow. The domain is the unit square $[0, 1]^2$, the radius of the circles is 0.15 and they are arranged in a cross form with a distance of 0.25 from the center of the domain as represented in Fig. 6. The velocity field is the following:

$$\boldsymbol{u}(x, y, t) = \begin{bmatrix} -2\sin^2(\pi x)\sin(\pi y)\cos(\pi y) \\ 2\sin^2(\pi y)\sin(\pi x)\cos(\pi x) \end{bmatrix} \cos\left(\pi \frac{t}{2}\right) \tag{17}$$

The mesh is a Cartesian grid composed of 128^2 cells. The total number of iteration is 4,000 and the time step is 5×10^{-4}. We use the Lagrangian remap algorithm with a Runge–Kutta 2 scheme to advect the fluids as presented in [1].

In this numerical test, more than two fluids can be present in one cell. To reconstruct the interface, we use the serial dissection [4] where the analytic reconstruction is used for the first fluid and the minimization algorithm is used for the remaining fluids. When four fluids are involved simultaneously in a cell, we use the B-tree dissection [4] to generate more combinations. As a consequence, the minimization algorithm is always involved in the reconstruction. To compare the minimization to the analytic reconstruction, we compare the time to reconstruct the interface with only the minimization algorithm for various tolerance values to the time to reconstruct the interface with both algorithms involved and for various tolerance values.

Table 2 presents the time ratio between minimization and analytic & minimization reconstruction for tolerance values of 10^{-15}, 10^{-12}, 10^{-9} and 10^{-6}. Note that we only measure the time of reconstruction.

We observe that the time ratio is always greater than 1 which means that the analytic reconstruction combined to the minimization is always faster that minimization alone. In the worst case, when we compare the analytic reconstruction combined to the minimization with a tolerance value of 10^{-15} to the minimization alone with a tolerance value of 10^{-6}, we observe that the proposed method is 33% faster than the classic minimization with a better precision. If we want to have the same precision with the classic minimization (10^{-15}), the time ratio becomes 2.67 for the benefit of the proposed method.

6. Conclusion

We have proposed an analytic MoF algorithm to reconstruct the interfaces on rectangular cells for two materials without minimization process. We have measured the CPU time of both the proposed algorithm and the minimization method on two test cases. The results show that the analytic reconstruction provides accurate results with a lower computational cost than the minimization method. When more than two materials are involved, the proposed algorithm can be applied only to the reconstruction of the first material, the remaining materials are reconstructed using the original minimization algorithm. Even in this case, we have shown that this algorithm is interesting in conjunction with the minimization algorithm.

Acknowledgements

This study was carried out with financial support from the French National Research Agency (ANR) in the framework of the "Investments for the future" Programme IdEx Bordeaux (ANR-10-IDEX-03-02), Cluster of excellence CPU.

The authors wish to thank the Aquitaine Regional Council for the financial support towards a 432-processor cluster investment, located in the I2M laboratory. Computer time for this study was also provided by the computing facilities MCIA (Mesocentre de Calcul Intensif Aquitain) of the Université de Bordeaux and of the Université de Pau et des Pays de l'Adour.

Appendix A. Flood algorithm for convex cells

Consider a convex polygon P composed of N vertices defined in counterclockwise order $\{\boldsymbol{p}_0, \cdots, \boldsymbol{p}_N\}$. The Flood Algorithm consists in finding a line segment Γ parametrized by its normal $\boldsymbol{n}$ and its distance to the origin $\xi^\star$ such that the part of the polygon behind the line segment matches a prescribed volume V, the flooding direction being given. The Algorithm 1 presents the Flood Algorithm we have implemented. The Fig. 7 illustrates the algorithm.

Algorithm 1: (Flood Algorithm) compute the distance $\xi^\star$ from the volume of fluid V and the flood direction $\boldsymbol{n}$.

```
input  : Convex polygon $P = \{\boldsymbol{p}_0, \cdots, \boldsymbol{p}_N\}$, volume of fluid $V$, flood direction $\boldsymbol{n}$
output : Distance $\xi^\star$

Find the point of the polygon with the minimal signed distance in direction $\boldsymbol{n}$ and call it $\boldsymbol{p}_0$.

$l \leftarrow 1$, $r \leftarrow N$;
$|\Gamma| \leftarrow 0$, $|\Gamma_{\text{next}}| \leftarrow 0$;
$\xi \leftarrow 0$, $\xi_{\text{next}} \leftarrow 0$;
$V_{\text{trapezoid}} \leftarrow 0$;

for $k \leftarrow 1$ to $N-1$ do
    if $\xi_r < \xi_l$ then
        $\xi_{\text{next}} \leftarrow \xi_r$;
        $r \leftarrow r + 1$;
    else
        $\xi_{\text{next}} \leftarrow \xi_l$;
        $l \leftarrow l - 1$;
    end
    $|\Gamma_{\text{next}}| \leftarrow \texttt{ComputeSection}(\xi_{\text{next}}, \boldsymbol{n}, P)$;
    $V_{\text{trapezoid}} \leftarrow 0.5(\xi_{\text{next}} - \xi)(|\Gamma_{\text{next}}| + |\Gamma|)$;
    if $V_{\text{tot}} + V_{\text{trapezoid}} \geq V$ then
        $\alpha \leftarrow \frac{V - V_{\text{tot}}}{V_{\text{trapezoid}}}$;
        $\beta \leftarrow \sqrt{\left(\frac{|\Gamma|}{|\Gamma_{\text{next}}| + |\Gamma|}\right)^2 + \alpha \frac{|\Gamma_{\text{next}}| - |\Gamma|}{|\Gamma_{\text{next}}| + |\Gamma|}} + \frac{|\Gamma|}{|\Gamma_{\text{next}}| + |\Gamma|}$;
        $\xi^\star \leftarrow \xi + (\xi_{\text{next}} - \xi)\frac{\alpha}{\beta}$;
        break;
    end
    $V_{\text{tot}} \leftarrow V_{\text{tot}} + V_{\text{trapezoid}}$;
    $\Gamma \leftarrow \Gamma_{\text{next}}$;
    $\xi \leftarrow \xi_{\text{next}}$;
end
```

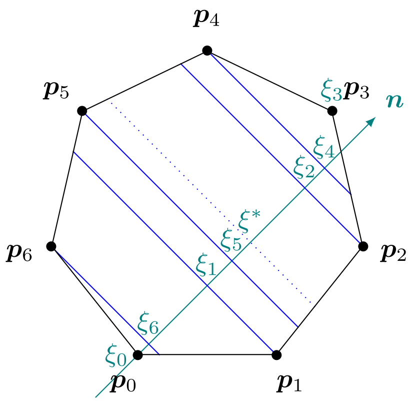

Fig. 7. Illustration of the Flood Algorithm on a convex polygon.

References

[1] V. Dyadechko, M. Shashkov, Moment-of-fluid interface reconstruction, Los Alamos National Laboratory Report LA-UR-05-7571.
[2] V. Dyadechko, M. Shashkov, Moment-of-fluid interface reconstruction, Los Alamos National Laboratory Report LA-UR-07-1537.
[3] H.T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, J. Comput. Phys. 226 (2) (2007) 2096–2132.
[4] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data, J. Comput. Phys. 227 (11) (2008) 5361–5384.
[5] H.T. Ahn, M. Shashkov, Adaptive moment-of-fluid method, J. Comput. Phys. 228 (8) (2009) 2792–2821.
[6] S. Galera, J. Breil, P.-H. Maire, A 2d unstructured multi-material cell-centered arbitrary lagrangian–eulerian (ccale) scheme using mof interface reconstruction, Comput. Fluids 46 (1) (2011) 237–244.
[7] S.P. Schofield, M.A. Christon, Effects of element order and interface reconstruction in fem/volume-of-fluid incompressible flow simulation, Int. J. Numer. Methods Fluids 68 (11) (2012) 1422–1437.
[8] J. Breil, T. Harribey, P.-H. Maire, M. Shashkov, A multi-material reale method with mof interface reconstruction, Comput. Fluids 83 (2013) 115–125.
[9] M. Jemison, E. Loch, M. Sussman, M. Shashkov, M. Arienti, M. Ohta, Y. Wang, A coupled level set-moment of fluid method for incompressible two-phase flows, J. Sci. Comput. 54 (2–3) (2013) 454–491.
[10] R.N. Hill, M. Shashkov, The symmetric moment-of-fluid interface reconstruction algorithm, J. Comput. Phys. 249 (2013) 180–184.
[11] M. Jemison, M. Sussman, M. Arienti, Compressible, multiphase semi-implicit method with moment of fluid interface representation, J. Comput. Phys. 279 (2014) 182–217.
[12] M. Jemison, M. Sussman, M. Shashkov, Filament capturing with the multimaterial moment-of-fluid method, J. Comput. Phys. 285 (2015) 149–172.
[13] D.L. Youngs, Time-dependent multi-material flow with large fluid distortion, Numer. Methods Fluid Dyn. 24 (1982) 273–285.
[14] J.E. Pilliod, E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, J. Comput. Phys. 199 (2) (2004) 465–502.
[15] H. Anbarlooei, K. Mazaheri, 'Moment of fluid' interface reconstruction method in axisymmetric coordinates, Int. J. Numer. Methods Biomed. Eng. 27 (10) (2011) 1640–1651.
[16] M.B. Friess, J. Breil, S. Galera, P.-H. Maire, M. Shashkov, A multi-material ccale-mof approach in cylindrical geometry, arXiv preprint, arXiv:1111.4900.
[17] P. Strobach, Solving cubics by polynomial fitting, J. Comput. Appl. Math. 235 (9) (2011) 3033–3052.
[18] P. Strobach, The fast quartic solver, J. Comput. Appl. Math. 234 (10) (2010) 3007–3024.
[19] J. Nocedal, S. Wright, Numerical Optimization, Springer Science & Business Media, 2006.
[20] J. Breil, S. Galera, P.-H. Maire, A two-dimensional vof interface reconstruction in a multi-material cell-centered ale scheme, Int. J. Numer. Methods Fluids 65 (11–12) (2011) 1351–1364.