INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2016; **81**:399–425
Published online 8 November 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/fld.4188

A mass-conserving level-set method for simulation of multiphase flow in geometrically complicated domains

F. Raees[1,2,*,†], D. R. van der Heul[1] and C. Vuik[1]

[1] *Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD, Delft, The Netherlands*
[2] *Department of Mathematics, NED University of Engineering and Technology, Karachi, Pakistan*

SUMMARY

The mass-conserving level-set (MCLS) method is a hybrid level-set (LS)/volume of fluid (VoF) based, interface capturing algorithm that combines the mass conserving properties of the VoF, with the benefits of having an explicit description of the interface of the LS method. The efficiency of the method is a result of the fact that the LS formulation allows evaluation of the VoF-field and VoF-fluxes without reconstruction of the interface in each cell. We present the extension of the MCLS method from its original formulation for Cartesian quadrilateral control volumes to triangular control volumes for optimal geometrical flexibility. The LS field is discretized using a second order discontinuous Galerkin method. After each time-step, a mass-conserving correction is imposed based on the simultaneously convected VoF field. This convection step is performed with a second-order Eulerian–Lagrangian approach, combined with a 'clipping' algorithm to project the advected field from the Lagrangian to the Eulerian grid. The MCLS method is shown to be accurately mass conserving and shows second order convergence for three different test cases. Copyright © 2015 John Wiley & Sons, Ltd.

Received 16 July 2015; Revised 7 October 2015; Accepted 11 October 2015

KEY WORDS: level-set; volume of fluid; mass-conserving; multiphase flow; discontinuous Galerkin

1. INTRODUCTION

Many industries, for example, oil and gas recovery, (nuclear) power generation, production of foods, and chemicals rely on stable and predictable liquid–gas multiphase flows for safe transport and processing. Nowadays, cost-effective system design and operation rely indispensably on (flow) simulation technology. Computational methods for analysis of immiscible two-phase flow in simple domains have reached a high level of maturity and technology readiness. However, many of these are not readily extended to be applied for more geometrically intricate domains, because they rely on interface reconstruction techniques that are only applicable for a discretization based on Cartesian control volumes. Algorithms that are applicable for triangular control volumes suffer from poor mass conservation and/or robustness because of their extreme complexity. Modeling the dynamics of two incompressible immiscible fluids is far more challenging than modeling single-phase flow because of the conflicting demands imposed by very strict mass conservation and accurately predicting the position of the interface between the phases, for which the smoothness and sharpness is preserved. The latter is especially important if the interface curvature and normal vector are required to model surface tension. Many different approaches have been proposed to mitigate the problems associated with modeling two-phase flow. Canonical examples are the marker and particle method [1], the volume of fluid (VoF) method of Hirt *et al.* [2], the level-set (LS) method

[*] Correspondence to: F. Raees, Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD, Delft, The Netherlands.

[†] E-mail: f.raees@tudelft.nl

of Osher and Sethian LS [3] and the front-tracking method [4]. Nearly all methods proposed in literature are directly based on one or a combination of these methods. Each has his advantages and disadvantages. Some methods are straightforward to implement because of their relative simplicity, but do not conserve mass while others are mass conserving but have high computational complexity, and can only be applied for the Cartesian control volumes they have been designed for. To handle interface movement without any geometrical restrictions, only two methods can be used: the VoF and the LS methods.

In the VoF method, a color function is used to identify the presence of either fluid. This color function has a value 1 in one fluid and 0 in the other fluid in order to distinguish between the two fluids. In the VoF method, the cell averaged value of the color function is used to compute the quantity of the fluid present in a cell. The interface location is only approximately defined: it will intersect those cells that have intermediate values of the VoF fraction between 0 and 1, but its exact location is unknown. In each time step, the position of the interface is reconstructed within the control volumes that are intersected by the interface, under the assumption that it is either aligned with one of the Cartesian coordinate directions [5] or simply planar [6], before it is advanced in time in a Lagrangian manner. Computation of the fluxes across the cell faces is performed using the concept of donor regions, leading to an algorithm that is mass conserving to machine precision.

In the LS method, the interface position is indicated by a zero LS of a function that is often chosen as a signed distance function. The LS is advected in time to get the position of the interface, using high-order discretization schemes for scalar advection equations. As the LS function is smoothly differentiable in the vicinity of the interface, geometric characteristics, like gradient and curvature are easy to obtain.

To keep the LS function a signed distance function, re-initialization needs to be performed. The VoF method is very accurately mass-conserving, but the interface reconstruction and the approximation of the local curvature of the interface to define surface tension effects are computationally intensive and involved, requiring filtering [2] or the construction of a height function. On the other hand, the LS approach provides an exact (up to discretization order) representation of the interface and straightforward evaluation of the local properties of the interface but is by definition not mass conserving: exact conservation of the LS function does not imply conservation of mass. In both cases, the curvature and normal vector can be computed with second order accuracy [7].

In recent years, an interest has developed in hybrid VoF/LS methods that combine the accurate mass conservation properties of the VoF method with the advantage of an exact representation of the interface by the LS method. The combined level-set volume of fluid method of Sussman *et al.* [8–10] and the mass-conserving level-set (MCLS) method of Van der Pijl *et al.* [11] are examples of this hybrid approach. Although both seem to be very similar on the surface, the major difference is that the combined level-set volume of fluid method basically combines the work involved in both the VoF and LS methods, while the MCLS method avoids the computationally intensive interface reconstruction step required for advection of the VoF field. This is accomplished by the use of the VoF function that directly relates the local value of the VoF field to the LS field and its gradient, without the need for an explicit reconstruction step.

The original MCLS method [11] was defined only for a discretization with (Cartesian) hexahedral (3D) and quadrilateral (2D) control volumes. Automatic tessellation of geometrically complex domains is generally accomplished using either triangular (2D) or tetrahedral (3D) control volumes. In this paper, the extension of MCLS to a discretization using unstructured (2D) triangular control volumes is presented. However, the ultimate aim is to extend MCLS method to general polygonal and polyhedral control volumes.

In this paper, we have presented the extension of the MCLS method toward unstructured 2D triangular grids. However, the proposed algorithm is in principle equally capable for the solution of 3D multiphase flow problems. We couple the LS field with the VoF field by means of analytic relations. Similar approaches can be found in [12, 13], but we believe our analytic relations are computationally more efficient and elegant. Furthermore, the efficiency of the algorithm is enhanced by using a discontinuous Galerkin discretization for the LS field. This is because the derived analytic relations are perfectly inline with the local nature of this type of discretization.

DOI: 10.1002/fld

The outline of this paper is as follows. First, we will briefly discuss the LS, the VoF, and the hybrid LS/VoF approach of the MCLS to modeling the interface between two immiscible incompressible fluids. Next the concept of the original MCLS is reviewed. We discuss all the constituents of the adaptation of the MCLS for triangular control volumes, followed by the derivation of the VoF function. After that, the algorithms for VoF and LS advection are discussed. The definition of the inverse of the VoF function to impose mass-conserving corrections on the LS field completes the description of the algorithm. The paper concludes with the results of a number of test cases obtained with the new algorithm.

2. MODELING THE EVOLUTION OF THE INTERFACE

The MCLS algorithm is used to model the dynamics of two-phase immiscible incompressible flow. It uses a 'hybrid' description of the distribution of the two-phases, $Phase_0$ and $Phase_1$, using both an LS and a VoF field.

2.1. *Level-set field*

The interface between the two phases $\boldsymbol{X}(t)$ is defined as an isoline of a C^2 function $l(\boldsymbol{x},t) : \Omega \rightarrow \mathbb{R}$, where $\Omega \subset \mathbb{R}^2$, and simply connected. Choosing $l(\boldsymbol{x},t)$ as the signed distance to the interface is one of the common choices. Consequently, this choice defines the interface as the zero-level contour of the signed distance function $\Phi(\boldsymbol{x},t)$. A signed distance function is a natural choice because it allows for accurate computation of the interface curvature and normal vector. However, Olssson *et al.* propose a LS function that resembles a mollified Heaviside function to improve the mass conservation properties [14–16]. In the MCLS, the LS function is chosen as a signed distance function $\Phi(\boldsymbol{x},t)$. If $\Phi(\boldsymbol{x},t) < 0, Phase_0$ is present, while $Phase_1$ is present when $\Phi(\boldsymbol{x},t) > 0$. Because of this implicit definition of the interface, complex topology changes, for example, merging of multiple or breaking up of a single interface are accommodated for. The smoothness of the LS function in the vicinity of the zero contour-line allows for a straightforward computation of geometrical quantities of the interface by evaluating derivatives of $\Phi(\boldsymbol{x},t)$ at the interface:

$$\hat{n}_\alpha(\boldsymbol{x},t) = \frac{\Phi_{,\alpha}(\boldsymbol{x},t)}{\sqrt{\Phi_{,\alpha}(\boldsymbol{x},t)\Phi_{,\alpha}(\boldsymbol{x},t)}}, \ \boldsymbol{x} \in \boldsymbol{X}(t), \quad \kappa(\boldsymbol{x},t) = \hat{n}_{\alpha,\alpha}(\boldsymbol{x},t), \ \boldsymbol{x} \in \boldsymbol{X}(t), \tag{1}$$

where $\hat{n}_\alpha$ is the unit normal vector and κ the curvature of a contour line of $\Phi(\boldsymbol{x},t) = 0$.

2.2. *Volume of fluid field*

The distribution of the fluids is represented by a C^{-1} color function $c(\boldsymbol{x},t) : \mathbb{R} \rightarrow \{0,1\}$: If $c(\boldsymbol{x},t) = 0, Phase_0$ is present, while $Phase_1$ is present when $c(\boldsymbol{x},t) = 1$. Based on the color function $c(\boldsymbol{x},t)$, we can define the Volume of Fluid *grid* function $\Psi_k : G \rightarrow [0,1]$ as

$$\Psi_k(t) = \frac{1}{|\Omega_k|}\int_{\Omega_k} c(\boldsymbol{x},t)d\Omega, \tag{2}$$

where G is the set of all centers of the control volumes Ω_k and $|\Omega_k|$ their area. The fact that $c(\boldsymbol{x},t)$ is C^{-1} continuous and $\Psi_k(t)$ only discretely defined makes the computation of the interface curvature challenging and computationally intensive.

2.3. *Hybrid formulation of the mass-conserving level-set method*

In the hybrid formulation, the flow is described by both an LS and a VoF field that are in agreement in each cell in the following way:

$$\Psi_k(t) = \frac{1}{|\Omega_k|}\int_{\Omega_k} H(\Phi(\boldsymbol{x},t))d\Omega, \tag{3}$$

where $H(q)$ is the Heaviside function. This means that the LS field has the mass-conserving properties of the VoF field, while at the same time the interface location is exactly defined by the LS field.

3. OVERVIEW OF THE MASS-CONSERVING LEVEL-SET ALGORITHM

In this section, an overview of the MCLS method is presented. The proposed algorithm relies on the key idea that a mass-conserving correction can be imposed on the LS field that is based on a simultaneously advected VoF field. The algorithm is visualized in Figure 1. Advancing the interface with the MCLS over a single time step consists of the following substeps, assuming the LS Φ^n and VoF field Ψ^n are known:

- The LS field and the VoF field are advected simultaneously. After this advection step, the two fields do not correspond to each other, that is, the fields do not comply with (3).
- The advected LS field Φ^* is used to compute a corresponding VoF field Ψ^* by means of the VoF function.
- The difference between the advected VoF field Ψ^{n+1} and the VoF field Ψ^* is determined.
- In an element, for which the difference is smaller than the prescribed tolerance, no correction to the LS field is required. If the difference exceeds the prescribed tolerance, a correction is imposed to the LS field by means of the inverse VoF function to make the LS field mass conserving up to a user specified tolerance.
- At time t^{n+1}, a mass-conserving LS field Φ^{n+1} has been obtained along with the VoF field Ψ^{n+1}.

In the following sections, each of the components that are required for the proposed extension of the MCLS algorithm for an unstructured triangular grid are presented. These are the LS field and

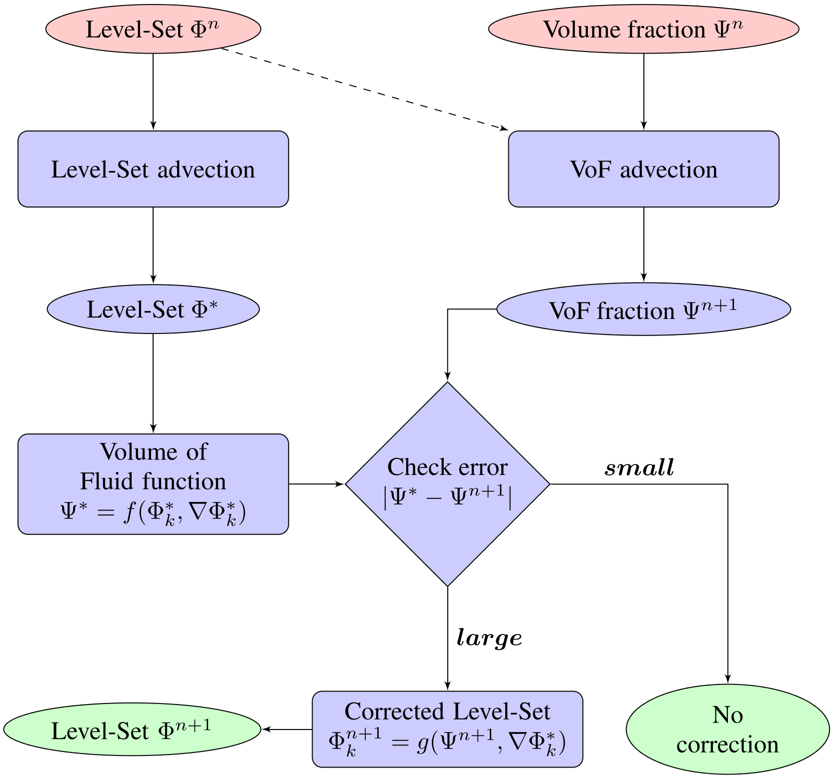

Level-Set Φ^n
Volume fraction Ψ^n
Level-Set advection
VoF advection
Level-Set Φ^*
VoF fraction Ψ^{n+1}
Volume of Fluid function $\Psi^* = f(\Phi_k^*, \nabla\Phi_k^*)$
Check error $|\Psi^* - \Psi^{n+1}|$
$small$
$large$
Level-Set Φ^{n+1}
Corrected Level-Set $\Phi_k^{n+1} = g(\Psi^{n+1}, \nabla\Phi_k^*)$
No correction

Figure 1. The mass-conserving level-set algorithm is presented in a flow chart to show the transfer of information between the different components involved in advancing a single time step.

DOI: 10.1002/fld

VoF field advection algorithms described in Sections 5 and 6, respectively, and the VoF function (Section 4) and its inverse (Section 7).

4. EFFICIENT COMPUTATION OF THE VOLUME OF FLUID FIELD FROM THE LEVEL-SET FIELD AND VICE VERSA

Consider the LS field $\Phi(\boldsymbol{x},t) : \Omega \to R$ and the VoF field $\Psi_k : G \to [0,1]$. Consider element Ω_k, with vertices $\boldsymbol{X}_k^{v1}$, $\boldsymbol{X}_k^{v2}$ and $\boldsymbol{X}_k^{v3}$, ordered in counter clock wise direction (Figure 2(a)). The centroid of the element is $\boldsymbol{X}_k$. If we choose $c(\boldsymbol{x},t) = H(\Phi(\boldsymbol{x},t))$, the VoF function in this element is given by (3). However, in a mixed cell, that is, a cell intersected by the interface, the VoF representation based on (3) is not computationally efficient. Therefore, (3) is approximated by a VoF function f:

$$\begin{aligned}\Psi_k(t) &= \frac{1}{|\Omega_k|}\int_{\Omega_k} H\left(\phi_k(\boldsymbol{x},t) + C_1 h^2\right) d\Omega \\ &= \frac{1}{|\Omega_k|}\int_{\Omega_k} H(\phi_k(\boldsymbol{x},t)) d\Omega + C_2 h^2 = f(\Phi_k, \nabla\Phi_k) + C_3 h^2, \quad C_i \in \mathbb{R},\end{aligned} \tag{4}$$

where $\phi_k(\boldsymbol{x},t)$ is the linearization of the LS function $(\Phi(\boldsymbol{x},t),\ \boldsymbol{x} \in \Omega_k)$, around the cell centroid $\boldsymbol{X}_k$, defined as

$$\phi_k(\boldsymbol{x},t) = \Phi_k(t) + \Phi_{,\alpha}\left(x_\alpha - (\boldsymbol{X}_k)_\alpha\right), \tag{5}$$

and where $\boldsymbol{X}_k$ is the coordinate vector of the cell centroid, $h = \sqrt{|\Omega_k|}$, $\Phi_k(t) = \Phi(\boldsymbol{X}_k, t)$ and $\Phi_{,\alpha}$ is the gradient of LS at cell centroid $\Phi_{,\alpha} = \nabla\Phi(\boldsymbol{X}_k)$. Now, the goal is to find the function $\Psi(\boldsymbol{X}_k) = f(\Phi_k, \nabla\Phi_k)$, such that it provides the VoF field in terms of the LS field.

4.1. *Converting the level-set field to the volume of fluid field*

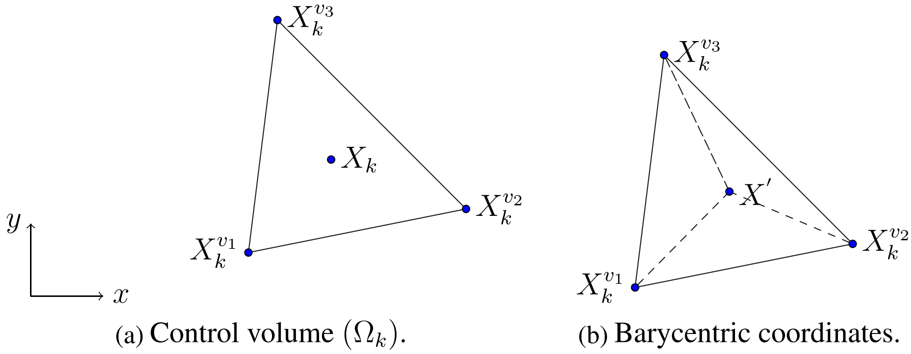
In order to express the VoF field in the LS field, the triangular control volumes are first mapped from Cartesian space to logical space using a barycentric coordinate transformation.

4.1.1. Barycentric transformation. Consider a triangular element Ω_k, as shown in Figure 2(b). This element can be divided into three triangles by assuming an arbitrary point $\boldsymbol{X}' \in \Omega_k$. This subdivision provides three triangles with areas $\blacktriangle \boldsymbol{X}_k^{v1}\boldsymbol{X}'\boldsymbol{X}_k^{v3}$, $\blacktriangle \boldsymbol{X}_k^{v1}\boldsymbol{X}_k^{v2}\boldsymbol{X}'$ and $\blacktriangle \boldsymbol{X}_k^{v2}\boldsymbol{X}_k^{v3}\boldsymbol{X}'$, respectively.

Barycentric coordinates ξ_β based on these areas, can be defined as

$$\xi_1 = \frac{\blacktriangle \boldsymbol{X}_k^{v1}\boldsymbol{X}'\boldsymbol{X}_k^{v3}}{|\Omega_k|}, \quad \xi_2 = \frac{\blacktriangle \boldsymbol{X}_k^{v1}\boldsymbol{X}_k^{v2}\boldsymbol{X}'}{|\Omega_k|}, \quad \xi_3 = \frac{\blacktriangle \boldsymbol{X}_k^{v2}\boldsymbol{X}_k^{v3}\boldsymbol{X}'}{|\Omega_k|}. \tag{6}$$

These barycentric coordinates define the following coordinate transformation on Ω_k:



(a) Control volume (Ω_k). (b) Barycentric coordinates.

Figure 2. Control volume (Ω_k), cell centroid (X_k), and arbitrary point (X').

DOI: 10.1002/fld

$$x_\alpha = \left(\boldsymbol{X}_k^{v\beta}\right)_\alpha \xi_\beta. \tag{7}$$

Naturally, for $\boldsymbol{x} \in \Omega_k$, the barycentric coordinates $\xi_\alpha \in [0, 1]$.

Using the fact that $\xi_1 + \xi_2 + \xi_3 = 1$, the dependency on one of the barycentric coordinates can be eliminated. For the moment, we will not decide on which of the coordinates is selected to be eliminated. Above transformation for the possible choices for the elimination of the dependency of either of the barycentric coordinates can be written as $\boldsymbol{x} = TC_i$, Where $T \in \mathbb{R}^{2\times3}$ defined as

$$T = \left[\, \boldsymbol{X}_k^{v1} \;\; \boldsymbol{X}_k^{v2} \;\; \boldsymbol{X}_k^{v3} \,\right],$$

and vector C_i represents the i^{th} column of $C \in \mathbb{R}^{3\times3}$, defined as

$$C = \begin{bmatrix} 1-\xi_2-\xi_3 & \xi_1 & \xi_1 \\ \xi_2 & 1-\xi_1-\xi_3 & \xi_2 \\ \xi_3 & \xi_3 & 1-\xi_1-\xi_2 \end{bmatrix},$$

where i is the index of the vertex of the triangular element. This information is necessary for the mapping of a specific vertex to the origin in logical space.

4.2. The volume of fluid function

Our aim is to use a geometric construction to find the function $\Psi_k = f(\Phi_k, \nabla\Phi_k)$. Contour lines of the linearized LS function $\phi_k(\boldsymbol{x})$ are straight line segments. The contour line $\phi_k(\boldsymbol{x}) = 0$ divides Ω_k in two pieces. From (3) and (4), it follows that $f(\phi_k, \nabla\phi_k)$ is the relative area of the polygon P bounded by the edges of the triangle and the line segment $\phi_k = 0$:

$$P = \{\boldsymbol{x} \in \Omega_k | \phi(\boldsymbol{x}) \leqslant 0\} \tag{8}$$

The polygon can be triangular, quadrilateral, or even reduced to a single point, depending on the intersection of the interface with Ω_k. Efficient evaluation of the function $f(\Phi_k, \nabla\Phi_k)$ requires a formulation that considers all possible situations as conveniently as possible.

For nonzero $|P|$, only two configurations are possible for the intersection of the interface with Ω_k. These possibilities are defined as follows:

- Case-1: $\Phi(\boldsymbol{X}_k^{v3}) < 0$ for a single vertex. In this case, P is a triangular as shown in Figure 3.
- Case-2: In this case, P is quadrilateral, which means two vertices are in the fluid of interest, that is, $\Phi(\boldsymbol{X}_k^{v3}) < 0$ and $\Phi(\boldsymbol{X}_k^{v2}) < 0$. This is shown in Figure 5.

However, it is sufficient to only consider Case-1, because in Case-2, we can use that $\Psi_k = f(\Phi_k, \nabla\Phi_k) = 1 - f(-\Phi_k, -\nabla\Phi_k)$. This means the relative area of the quadrilateral can be

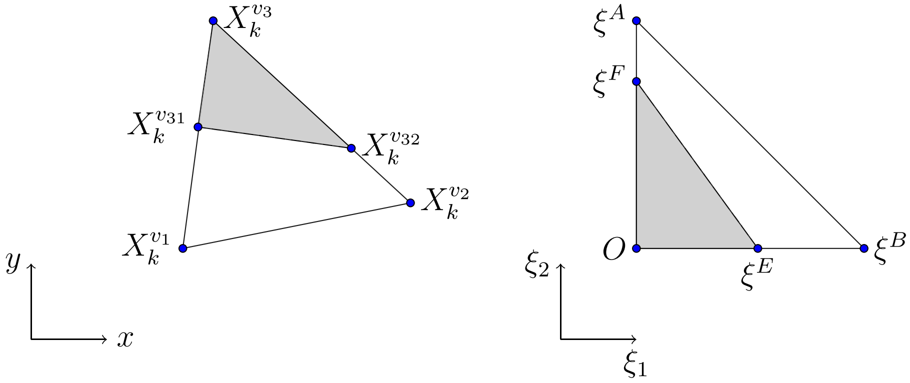


Figure 3. Case-1: Triangular domain of interest, where the vertex $\boldsymbol{X}_k^{v3}$ is mapped to the origin.

obtained by subtracting the relative area of the complementary part of the triangle. It is important to point out that this is not the only option to handle Case-2. It is indeed possible to define a single VoF function that can handle both cases.

4.2.1. Case-1: triangular domain of interest. Firstly, we consider Case-1. The vertex $\boldsymbol{X}_k^{v3}$ is selected to be mapped to the origin, as is shown in Figure 3. The linear transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is bijective for a non degenerate triangle Ω_k: a triangle for which no vertices coincide and/or edges are aligned. Define the (nonsingular) Jacobian of this mapping as $J = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}$. The linearized LS function $\tilde{\phi}(\boldsymbol{\xi})$ can now be formulated in logical space as:

$$\tilde{\phi}(\boldsymbol{\xi}, t) = \Phi(\boldsymbol{x}(\boldsymbol{\xi}_k), t) + \left.\frac{\partial \Phi}{\partial \boldsymbol{\xi}}\right|_{(\boldsymbol{\xi},t)=(\boldsymbol{\xi}_k,t)} (\boldsymbol{\xi} - \boldsymbol{\xi}_{\boldsymbol{k}}), \quad \frac{\partial \Phi}{\partial \boldsymbol{\xi}} = J \nabla \Phi, \tag{9}$$

where $\underline{\boldsymbol{\xi}_k = \boldsymbol{\xi}}(\boldsymbol{X}_k)$. Consider Figure 3 that shows the interface is intersecting the edges $\overline{\boldsymbol{X}_k^{v3}\boldsymbol{X}_k^{v1}}$ and $\overline{\boldsymbol{X}_k^{v2}\boldsymbol{X}_k^{v3}}$ at the points $\boldsymbol{X}_k^{v31}$ and $\boldsymbol{X}_k^{v32}$, respectively. The vertex $\boldsymbol{X}_k^{v3}$, which is common to the two edges intersected by the interface, is mapped to the origin while the other two vertices are mapped to points on the axes in logical space. It is important to note that vertices $\boldsymbol{X}_k^{v1}$ and $\boldsymbol{X}_k^{v2}$ can be mapped to any of the vertices in logical space, except for the origin. The other two vertices in logical space are located at $(1,0)^T$ and $(0,1)^T$. Furthermore, the interface is represented by a line segment $\overline{\boldsymbol{X}_k^{v31}\boldsymbol{X}_k^{v32}}$. Our target is to compute the area of the sub triangle $\blacktriangle \boldsymbol{X}_k^{v31}\boldsymbol{X}_k^{v3}\boldsymbol{X}_k^{v32}$ with the help of the linearized LS field derived in Equation (9).

Let us assume that the vertices $\boldsymbol{X}_k^{v1}$ and $\boldsymbol{X}_k^{v2}$ are mapped to $\boldsymbol{\xi} = (1,0)^T$ and $\boldsymbol{\xi} = (0,1)^T$ respectively, in logical space. Therefore, points $\boldsymbol{X}_k^{v31}$ and $\boldsymbol{X}_k^{v32}$ are mapped to $\boldsymbol{\xi}^E = (\xi_1^E, 0)^T$ and $\boldsymbol{\xi}^F = (0, \xi_2^F)^T$, respectively. This is shown in Figure 3. In order to compute the area of the region $\boldsymbol{X}_k^{v31}\boldsymbol{X}_k^{v3}\boldsymbol{X}_k^{v32}$, the images of the points E and F are needed. In the next subsection expressions are derived for $\boldsymbol{\xi}^E$ and $\boldsymbol{\xi}^F$.

4.2.2. Coordinates of the point $\boldsymbol{\xi}^E$. The logical space coordinates of the cell centroid $\boldsymbol{\xi}_k$ are by virtue of the barycentric coordinate transformation always given by $\boldsymbol{\xi}_k = \left(\frac{1}{3}, \frac{1}{3}\right)^T$. This means (9) can be written as (dropping the explicit dependence on t)

$$\tilde{\phi}(\boldsymbol{\xi}) = \Phi_k + \frac{\partial \Phi}{\partial \xi_1}\left(\xi_1 - \frac{1}{3}\right) + \frac{\partial \Phi}{\partial \xi_2}\left(\xi_2 - \frac{1}{3}\right). \tag{10}$$

Substitution of $\boldsymbol{\xi}^E$ in (10) leads to

$$\tilde{\phi}(\boldsymbol{\xi}^E) = \Phi_k + \frac{\partial \Phi}{\partial \xi_1}\left(\xi_1^E - \frac{1}{3}\right) - \frac{\partial \Phi}{\partial \xi_2}\left(\frac{1}{3}\right) = 0. \tag{11}$$

The right-hand side of (11) is zero, because the interface passes through the point $\boldsymbol{\xi}^E$. Evaluation of (10) in $\boldsymbol{\xi}(\boldsymbol{X}_k^{v3}) = \boldsymbol{0}$ gives

$$\tilde{\phi}(\boldsymbol{0}) = \Phi_k - \frac{\partial \Phi}{\partial \xi_1}\left(\frac{1}{3}\right) - \frac{\partial \Phi}{\partial \xi_2}\left(\frac{1}{3}\right), \tag{12}$$

where $\tilde{\phi}(\boldsymbol{0})$ is the value of the linearized LS field at the origin in logical space. Solving (11) and (12) to find ξ_1^E leads to

$$\xi_1^E = \left| -\frac{\tilde{\phi}(\boldsymbol{0})}{D_{\xi_1}} \right|, \tag{13}$$

DOI: 10.1002/fld

where $D_{\xi_1} = \dfrac{\partial \Phi}{\partial \xi_1}$. Therefore, the logical space coordinates of the point E are given by $\boldsymbol{\xi}^E = \left(\dfrac{\tilde{\phi}(\mathbf{0})}{D_{\xi_1}}, 0 \right)$.

Using a similar procedure in the ξ_2 direction, the coordinates of the point $\boldsymbol{\xi}^F$ in logical space can be found as $\boldsymbol{\xi}^F = \left(0, \dfrac{\tilde{\phi}(\mathbf{0})}{D_{\xi_2}} \right)$ and $D_{\xi_2} = \dfrac{\partial \Phi}{\partial \xi_2}$.

4.2.3. Evaluation of the VoF from the LS function. Now, we can compute the area enclosed by the points $\mathbf{0}, \boldsymbol{\xi}^E$, and $\boldsymbol{\xi}^F$ in logical space. The area of the enclosed region is denoted by $A_{\boldsymbol{\xi}}$ and is given by

$$A_{\boldsymbol{\xi}} = \frac{(\tilde{\phi}(\mathbf{0}))^2}{2 D_{\xi_1} D_{\xi_2}}. \tag{14}$$

Equation (14) represents the area of the region enclosed by line segments that connect $\mathbf{0}, \boldsymbol{\xi}^E$, and $\boldsymbol{\xi}^F$ in logical space. The product of $A_{\boldsymbol{\xi}}$ and the Jacobian of the transformation J is equal to the area enclosed by the points $\boldsymbol{X}_k^{v31} \boldsymbol{X}_k^{v3} \boldsymbol{X}_k^{v32}$ in physical space.

The area of the image of Ω_k in logical space, $A_{\boldsymbol{\xi}}^{\text{total}}$, is $\frac{1}{2}$. The VoF in both physical and logical space can be defined as

$$\Psi(\boldsymbol{x}_k) = \frac{A_{\boldsymbol{\xi}}}{A_{\boldsymbol{\xi}}^{\text{total}}} = \frac{(\tilde{\phi}(\mathbf{0}))^2}{D_{\xi_1} D_{\xi_2}}. \tag{15}$$

This is the VoF function that returns the VoF value in each triangular element when the value of the linearized LS field is known at the cell center. However, (15) is only valid when the domain of interest is triangular. This means that it is not valid when the interface intersects or passes above $\boldsymbol{\xi}^A$. The domain of (15) can be derived by considering the limit case of the interface intersecting $\boldsymbol{\xi}^A$. The linearized LS at the origin is given by

$$\tilde{\phi}(\mathbf{0}) = \Phi_k - \frac{D_{\xi_1}}{3} - \frac{D_{\xi_2}}{3}. \tag{16}$$

This can be written as

$$\begin{aligned} \tilde{\phi}(\mathbf{0}) &= \Phi_k - \frac{D_{\xi_1}}{3} + \frac{2D_{\xi_2}}{3} - D_{\xi_2} \Leftrightarrow \\ \tilde{\phi}(\mathbf{0}) &= \tilde{\phi}(\boldsymbol{\xi}^{\boldsymbol{F}}) - D_{\xi_2}. \end{aligned}$$

Under the assumption that the interface is passing through the vertex $\boldsymbol{\xi}^{\boldsymbol{A}}, \tilde{\phi}(\boldsymbol{\xi}^{\boldsymbol{F}}) = 0$ by definition and

$$\tilde{\phi}(\mathbf{0}) = -D_{\xi_2}. \tag{17}$$

This means that for all the values of $\tilde{\phi}(\mathbf{0}) \geqslant -D_{\xi_2}$ (15) holds. Furthermore, the assumptions on the mapping to logical space require $\tilde{\phi}(\mathbf{0}) \geqslant 0$ and

$$\Psi(x_k) = \frac{(\tilde{\phi}(\mathbf{0}))^2}{D_{\xi_1} D_{\xi_2}}, \qquad -D_{\xi_2} \leqslant \tilde{\phi}(\mathbf{0}) \leqslant 0. \tag{18}$$

The range of the VoF function can now be defined and the complete definition of the VoF function for Case-1 is given as

$$\Psi : [-D_{\xi_2}, 0] \times \mathbb{R}^2 \to \left[0, \frac{D_{\xi_2}}{D_{\xi_1}} \right], \quad \Psi(\tilde{\phi}(\mathbf{0}), \boldsymbol{D}_{\boldsymbol{\xi}}) = \frac{(\tilde{\phi}(\mathbf{0}))^2}{D_{\xi_1} D_{\xi_2}}. \tag{19}$$

Note that the definition of the mapping implies $D_{\xi_1} D_{\xi_2} \geqslant 0$.

If the interface passes through one of the vertices, (15) is valid. This is illustrated in Figure 4 where the interface is passing through vertex $\boldsymbol{X}_k^{v1}$. In this case, vertex $\boldsymbol{X}_k^{v3}$ is selected to be mapped to the origin and vertex $\boldsymbol{X}_k^{v1}$ is mapped to the ξ_2 axis, as it has the lower magnitude of the LS field of the two other vertices, that is, $\Phi(\boldsymbol{X}_k^{v_1}) = 0$.

If the interface is exactly aligned with one of the edges of the triangular element, Ψ_k is either 0 or 1, depending on the fluid of interest. In this case, there is no need to use (19). The definition of the mapping excludes the possibility that the interface is aligned with one of the coordinate axes in logical space for Case-1. However, for Case-2, it is possible that the interface ends up aligned with one of the axes in the logical space.

4.2.4. Case-2: quadrilateral domain of interest. An example of Case-2 is presented in Figure 5, where vertices $\Phi(\boldsymbol{X}_k^{v3}) < 0$ and $\Phi(\boldsymbol{X}_k^{v2}) < 0$ are now both located in the domain of interest. We assume vertex $\boldsymbol{X}_k^{v2}$ has a lower magnitude of the LS as compared with vertex $\boldsymbol{X}_k^{v3}$. Therefore, vertex $\boldsymbol{X}_k^{v2}$ is mapped to the ξ_2 axis and $\boldsymbol{X}_k^{v3}$ to the origin in logical space. The fluid of interest is enclosed by the quadrilateral $\boldsymbol{0}$, $\boldsymbol{\xi}^E$, $\boldsymbol{\xi}^I$, $\boldsymbol{\xi}^A$ (Figure 5). The vertex $\boldsymbol{\xi}^F$ is located outside the mapped region and intersects the ξ_2 axis, provided that the interface is not parallel to one of the edges of the triangle. This will only occur when the interface is exactly equidistant from the vertices $\boldsymbol{X}_k^{v3}$ and $\boldsymbol{X}_k^{v2}$. This case will be discussed separately in Section 4.2.5.

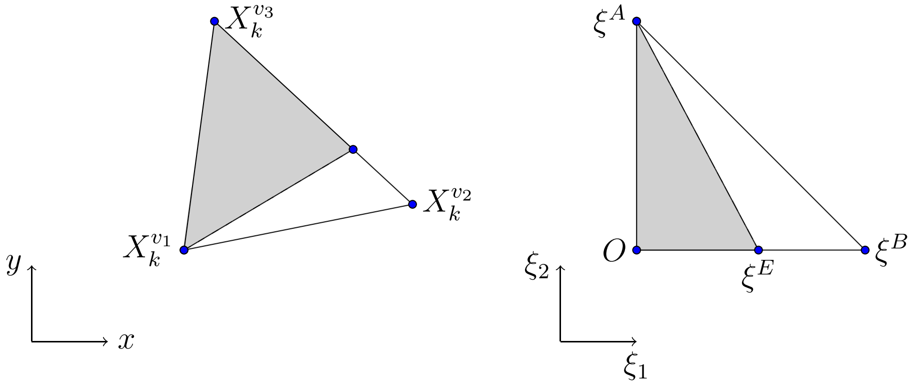


Figure 4. Interface passing through vertex ($X_k^{v_1}$).

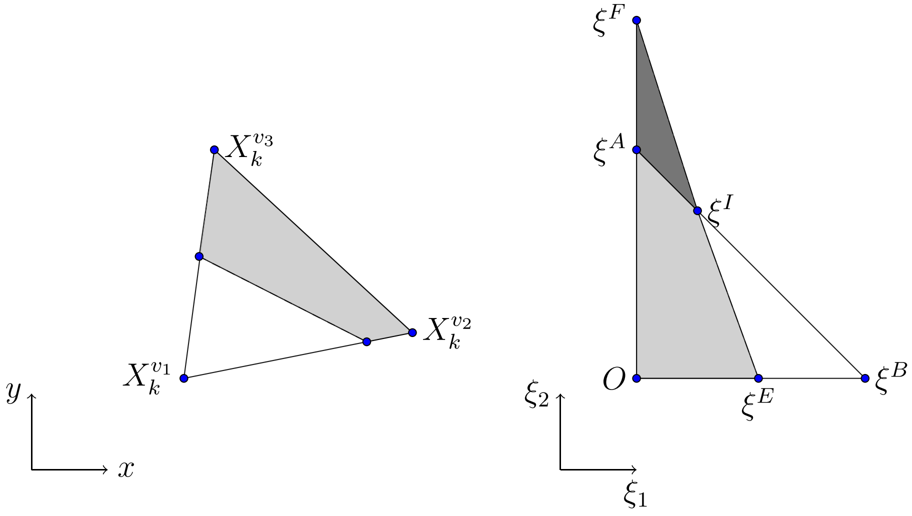


Figure 5. Case-2: mapping of two nodes a side.

DOI: 10.1002/fld

We can now use (14) to compute the area of the triangular region enclosed by vertices $\mathbf{0}, \boldsymbol{\xi}^E, \boldsymbol{\xi}^I, \boldsymbol{\xi}^F$ and $\boldsymbol{\xi}^A$. Let us call this area $\boldsymbol{A_{tri.}}$. To find the area of the quadrilateral region $\boldsymbol{A_{\xi}}$, the area of the triangular region defined by the vertices $\boldsymbol{\xi}^F, \boldsymbol{\xi}^I, \boldsymbol{\xi}^A$ that we will refer to as $\boldsymbol{A_{ex.}}$, needs to be subtracted from the area $\boldsymbol{A_{tri.}}$:

$$\boldsymbol{A_{\xi}} = \frac{(\tilde{\phi}(\mathbf{0}))^2}{2D_{\xi_1}D_{\xi_2}} - \boldsymbol{A_{ex.}}. \tag{20}$$

In order to compute $\boldsymbol{A_{ex.}}$, the coordinates of the point $\boldsymbol{\xi}^I$ are required. This point is the intersection of the interface and the edge of the triangle opposing the origin of the reference triangle. At $\boldsymbol{\xi}^I$, we have $\Phi(\boldsymbol{\xi}^I) = 0$. The linearized LS at this point is defined as

$$\phi(\boldsymbol{\xi}^I) = \Phi_k + D_{\xi_1}\left(\xi_1^I - \frac{1}{3}\right) + D_{\xi_2}\left(\xi_2^I - \frac{1}{3}\right) = 0. \tag{21}$$

At the edge $\overline{\boldsymbol{\xi}^A\boldsymbol{\xi}^B}$, we have $\xi_2^I = 1 - \xi_1^I$, using this relation in (21) leads to

$$\xi_1^I = -\frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})}{D_{\xi_1} - D_{\xi_2}}, \quad \xi_2^I = \frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_1})}{D_{\xi_1} - D_{\xi_2}}, \tag{22}$$

where $\tilde{\phi}(\mathbf{0}) = \Phi_k - \frac{D_{\xi_1}}{3} - \frac{D_{\xi_2}}{3}$. The coordinates of the other two points that define the quadrilateral area are $\boldsymbol{\xi}^F = \boldsymbol{\xi}^F\left(0, \frac{-\tilde{\phi}(\mathbf{0})}{D_{\xi_2}}\right)$ and $\boldsymbol{\xi}^A = \boldsymbol{\xi}^A(0,1)$. Define the following vectors: $\overrightarrow{\boldsymbol{\xi}^F\boldsymbol{\xi}^A}$ and $\overrightarrow{\boldsymbol{\xi}^F\boldsymbol{\xi}^I}$:

$$\begin{aligned}
\overrightarrow{\boldsymbol{\xi}^F\boldsymbol{\xi}^A} &= \left(0, 1 + \frac{\tilde{\phi}(\mathbf{0})}{D_{\xi_2}}\right)^T, \\
\overrightarrow{\boldsymbol{\xi}^F\boldsymbol{\xi}^I} &= \left(-\frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})}{D_{\xi_1} - D_{\xi_2}}, \frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_1})}{D_{\xi_1} - D_{\xi_2}} + \frac{\tilde{\phi}(\mathbf{0})}{D_{\xi_2}}\right)^T.
\end{aligned} \tag{23}$$

Then the excess area $\boldsymbol{A_{ex.}}$ is defined as

$$\boldsymbol{A_{ex.}} = \frac{\left\|\left[\overrightarrow{\boldsymbol{\xi}^F\boldsymbol{\xi}^A} \times \overrightarrow{\boldsymbol{\xi}^F\boldsymbol{\xi}^I}\right]\right\|}{2} = \frac{\left(\tilde{\phi}(\mathbf{0}) + D_{\xi_2}\right)^2}{2D_{\xi_2}\left(D_{\xi_1} - D_{\xi_2}\right)}. \tag{24}$$

Substitution of (24) in (20) leads to

$$\boldsymbol{A_{\xi}} = \frac{(\tilde{\phi}(\mathbf{0}))^2}{2D_{\xi_1}D_{\xi_2}} - \frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})^2}{2D_{\xi_2}\left(D_{\xi_1} - D_{\xi_2}\right)}. \tag{25}$$

The VoF function for Case-2 is now given by

$$\Psi_k = \frac{(\tilde{\phi}(\mathbf{0}))^2}{D_{\xi_1}D_{\xi_2}} - \frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})^2}{D_{\xi_2}(D_{\xi_1} - D_{\xi_2})}. \tag{26}$$

Note that $(D_{\xi_1} - D_{\xi_2})$ cannot be zero, because this will only occur when the interface is parallel to the edge $\overline{\boldsymbol{\xi}^A\boldsymbol{\xi}^B}$. The domain of the VoF function (26) needs to be defined such that the range of the function is a subset of the unit interval. In order to derive an upper bound, consider Figure 5. Because of the definition of the mapping $|\tilde{\phi}(\boldsymbol{\xi}^A)| < |\tilde{\phi}(\boldsymbol{\xi}^B)|$. Therefore, the interface will pass outside the element when $\tilde{\phi}(\mathbf{0}) > -D_{\xi_1}$. The value of $\tilde{\phi}(\mathbf{0}) = -D_{\xi_1}$ corresponds to $\Psi_k = 1$. Therefore, the domain of the VoF function (26) is equal to the interval $[-D_{\xi_1}, -D_{\xi_2}]$ and

$$\Psi_k = \frac{(\tilde{\phi}(\mathbf{0}))^2}{D_{\xi_1} D_{\xi_2}} - \frac{\left(\tilde{\phi}(\mathbf{0}) + D_{\xi_2}\right)^2}{D_{\xi_2}\left(D_{\xi_1} - D_{\xi_2}\right)} \quad \forall \quad -D_{\xi_1} \leqslant \tilde{\phi}(\mathbf{0}) \leqslant -D_{\xi_2}. \tag{27}$$

Given the domain of the VoF function, its range can be derived. The complete definition of the VoF function for a quadrilateral domain of interest is given as

$$\begin{aligned} &\Psi : \left[-D_{\xi_1}, -D_{\xi_2}\right] \times \mathbb{R}^2 \to \left[\frac{D_{\xi_2}}{D_{\xi_1}}, 1\right], \\ &\Psi(\tilde{\phi}(\mathbf{0}), \boldsymbol{D}_{\boldsymbol{\xi}}) = \frac{(\tilde{\phi}(\mathbf{0}))^2}{D_{\xi_1} D_{\xi_2}} - \frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})^2}{D_{\xi_2}(D_{\xi_1} - D_{\xi_2})}. \end{aligned} \tag{28}$$

4.2.5. When the interface is parallel to one of the edges. Equation (28) is derived for the case when the interface is not parallel to the edge $\overline{O\boldsymbol{\xi}^{\boldsymbol{A}}}$. The latter occurs when the interface is equidistant from both vertices $\boldsymbol{O}$ and $\boldsymbol{\xi}^{\boldsymbol{A}}$. In this case, the extra area would become unbounded, so Equation (26) can not be used. In this situation the domain of interest is a trapezoid as shown in Figure 6 and its area can be computed easily as

$$\boldsymbol{A}_{\boldsymbol{\xi}} = \frac{-\tilde{\phi}(\mathbf{0})(\tilde{\phi}(\mathbf{0}) + 2D_{\xi_1})}{2}, \tag{29}$$

and the VoF for this case is given as

$$\Psi_k = -\tilde{\phi}(\mathbf{0})(\tilde{\phi}(\mathbf{0}) + 2D_{\xi_1}) \tag{30}$$

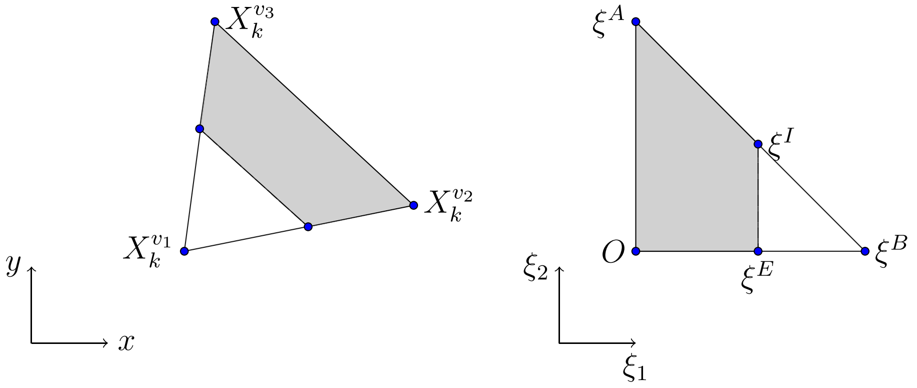


Figure 6. Mapping of the interface parallel to one of its edge.

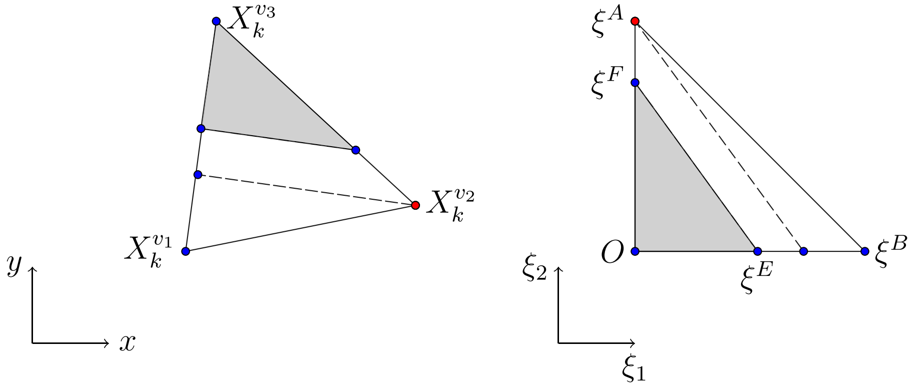


Figure 7. Combination of two cases.

DOI: 10.1002/fld

4.3. Combining the volume of fluid functions for Case-1 and Case-2

The union of the domains of the two VoF functions for Case-1 and Case-2 together covers the set $\left[-D_{\xi_1}, 0\right]$ of possible values of $\tilde{\phi}(\mathbf{0})$, under the assumption that the control volume of interest is intersected by the interface, see Figure 7. Both VoF functions have a quadratic dependence on $\tilde{\phi}(\mathbf{0})$ and are C^1 continuous at the common point $\tilde{\phi}(\mathbf{0}) = -D_{\xi_2}$. This is shown in Figure 8. The range of the VoF function should be [0, 1].

In order to select correct branch of the two VoF functions, we define a switching parameter $S_p(\tilde{\phi}(\mathbf{0}))$ as

$$S_p(\tilde{\phi}(\mathbf{0})) = \begin{cases} 0 & : \tilde{\phi}(\mathbf{0}) \geqslant -D_{\xi_2} \\ 1 & : \tilde{\phi}(\mathbf{0}) < -D_{\xi_2} \end{cases}. \tag{31}$$

Using (31), (19), and (28) can be combined to a single VoF function as

$$\begin{aligned} &\Psi : \left[-D_{\xi_1}, 0\right] \times \mathbb{R}^2 \to [0,1], \\ &\Psi(\tilde{\phi}(\mathbf{0}), \boldsymbol{D_\xi}) = \frac{(\tilde{\phi}(\mathbf{0}))^2}{D_{\xi_1} D_{\xi_2}} - S_p(\tilde{\phi}(\mathbf{0})) \frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})^2}{D_{\xi_2}\left(D_{\xi_1} - D_{\xi_2}\right)}. \end{aligned} \tag{32}$$

Let us further define two coefficients:

$$c_1 = \frac{1}{D_{\xi_1} D_{\xi_2}}, \qquad c_2 = \frac{-S_p(\tilde{\phi}(\mathbf{0}))}{D_{\xi_2}\left(D_{\xi_1} - D_{\xi_2}\right)}. \tag{33}$$

Using (33) and (32) a uniformly valid expression for Ψ_k can be formulated as

$$\Psi_k = (c_1 + c_2)(\tilde{\phi}(\mathbf{0}))^2 + 2c_2 D_{\xi_2}(\tilde{\phi}(\mathbf{0})) + c_2 D_{\xi_2}^2 \ \forall \quad -D_{\xi_1} \leqslant \tilde{\phi}(\mathbf{0}) \leqslant 0. \tag{34}$$

It is important to note that (34) also holds for the case when the interface is aligned with or parallel to the edge $\overline{O\boldsymbol{\xi}^{\boldsymbol{A}}}$, that is, $D_{\xi_2} = 0$. In Case-1, that is, $S_p(\tilde{\phi}(\mathbf{0})) = 0$, the interface can be aligned with edge $\overline{O\boldsymbol{\xi}^{\boldsymbol{A}}}$ and in this situation the VoF is either 0 or 1. In Case-2, that is, $S_p(\tilde{\phi}(\mathbf{0})) = 1$, the interface can be parallel to the edge $\overline{O\boldsymbol{\xi}^{\boldsymbol{A}}}$, but the behavior of the coefficients $c_1 + c_2$ and $D_{\xi_2} c_2$ in the polynomial is not singular when $D_{\xi_2} = 0$.

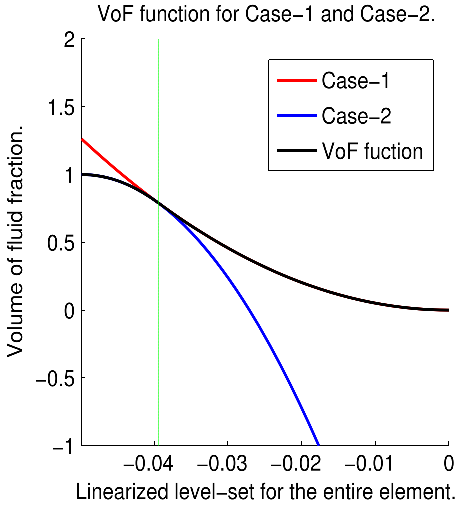


Figure 8. VoF function for both Case-1 and Case-2.

DOI: 10.1002/fld

The VoF function (34) is expressed in the value of the linearized LS field at the origin in logical space and the partial derivatives of the LS function with respect to the logical space coordinates. However, we have defined the VoF function (4) in terms of the cell centroid value and gradient of the LS. To formulate the VoF function explicitly in the latter form, (16) is used in (34) to find the following expression for $f(\Phi_k, \nabla\Phi_k)$:

$$
\begin{aligned}
\Psi_k &= f(\Phi_k, \nabla\Phi_k) = \left[\frac{-2D_{\xi_1} + D_{\xi_2}}{3}, \frac{D_{\xi_1} + D_{\xi_2}}{3}\right] \to [0,1],\\
f(\Phi_k, \nabla\Phi_k) &= (c_1 + c_2)\Phi_k^2 + \left(\frac{(c_1 + c_2)D_{\xi_1}^2 + (2c_1 - 4c_2)D_{\xi_1}D_{\xi_2} + (c_1 + 4c_2)D_{\xi_2}^2}{9}\right)\Phi_k\\
&\quad - \frac{2\left((c_1 + c_2)D_{\xi_1} + (c_1 - 2c_2)D_{\xi_2}\right)}{3}, \quad [D_{\xi_1} \;\; D_{\xi_2}]^T = J\nabla\Phi_k.
\end{aligned}
\tag{35}
$$

5. ADVECTION OF THE LEVEL-SET FIELD

Because the interface is by definition a contour line of the LS function, the following equation holds at the interface:

$$
\frac{d}{dt}\Phi(\boldsymbol{x}, t) = 0 \Rightarrow \frac{\partial\Phi(\boldsymbol{x}, t)}{\partial t} + u_\alpha(\boldsymbol{x}, t)\Phi(\boldsymbol{x}, t)_{,\alpha} = 0, \; \boldsymbol{x} \in \boldsymbol{X}(t), \; t > 0. \tag{36}
$$

In the MCLS method, Equation (36) is postulated to hold for all $\boldsymbol{x} \in \Omega$, but other choices are possible, as long as they are consistent with (36) and lead to a function that is at least C^2 continuous in the vicinity of the interface to allow computation of the curvature. For a solenodial velocity field $u_\alpha(\boldsymbol{x}, t)$, the extension of (36) is given by (omitting the explicit dependence of $\boldsymbol{x}$ ant t of $u_\alpha(\boldsymbol{x}, t)$)

$$
\frac{\partial\Phi(\boldsymbol{x}, t)}{\partial t} + (u_\alpha\Phi(\boldsymbol{x}, t))_{,\alpha} = 0, \; \boldsymbol{x} \in \Omega, \; t > 0. \tag{37}
$$

This equation shows that the LS function is conserved. However, a conservative redistribution of the LS function can lead to a change in the area enclosed by the interface. This is one of the drawbacks associated with the LS method. To some extent, this can be remedied by using higher order approximations of the convection operator, for example, using ENO or WENO schemes in the context of a finite volume discretization [17, 18], combined with higher order time-integration methods or by applying adaptive grid refinement near the interface. However, all the remedies are not computationally efficient when (37) is discretized on an unstructured set of triangular control volumes. In the current paper, a higher-order discontinuous Galerkin (DG) finite element method is used for the discretization of (37). This approach leads to a high order of accuracy boundary conforming discretization for domains with arbitrary geometrical complexity. For integration in time, a low storage Runge–Kutta method is used. A brief description of the spatial and temporal discretization of (37) is presented, based on [19–22]. The computational domain Ω is discretized into K non-overlapping, straight-sided triangular elements Ω_k. Lagrange polynomials $L_i(\boldsymbol{x})$ of degree N are used as basis functions to expand the solution in element Ω_k as

$$
\Phi_k^h(\boldsymbol{x}, t) = \Sigma_{i=1}^m \phi_i^k(t)L_i(\boldsymbol{x}), \; \boldsymbol{x} \in \Omega_k, \; t > 0, \tag{38}
$$

where m is the total number of nodal points in an element. In the DG frame work, the residual is made orthogonal to the polynomial space in Ω_k, by requiring (omitting the explicit dependence of $\boldsymbol{x}$ ant t of $\Phi_k^h(\boldsymbol{x}, t)$)

$$
\int_{\Omega_k} \left(\frac{\partial\Phi_k^h}{\partial t} + \left(u_\alpha\Phi_k^h\right)_{,\alpha}\right) L_i(\boldsymbol{x}) d\Omega = 0, \; i = 1 \ldots m. \tag{39}
$$

DOI: 10.1002/fld

The weak form of Equation (37) can be obtained by applying integration by parts:

$$\int_{\Omega_k} \frac{\partial \Phi_k^h}{\partial t} L_i(\boldsymbol{x}) - L_{i,\alpha}(\boldsymbol{x}) u_\alpha \Phi_k^h d\Omega = -\oint_{\partial\Omega_k} (\hat{n}_\alpha u_\alpha \Phi^h)^* L_i(\boldsymbol{x}) d\Omega, \ i = 1 \dots m, \tag{40}$$

where $\hat{n}_\alpha$ is the outward pointing unit normal and $(u_\alpha \Phi^h)^*$ is the numerical flux, used to impose boundary conditions on each element. Integrating by parts once again leads to the strong formulation of the DG method:

$$\int_{\Omega_k} \left(\frac{\partial \Phi^h}{\partial t} + (u_\alpha \Phi^h)_{,\alpha} \right) L_i(\boldsymbol{x}) d\Omega = \oint_{\partial\Omega_k} n_\alpha \left(u_\alpha \Phi^h - \left(u_\alpha \Phi^h \right)^* \right) L_i(\boldsymbol{x}) d\Omega, \ i = 1 \dots m. \tag{41}$$

In the MCLS, (41) is combined with the Lax–Friedrichs approximation of the numerical flux [20]. The system ((41)) is formulated for all control volumes Ω_k, $k = 1 \dots K$. All equations are coupled through the numerical flux function and advanced simultaneously.

6. ADVECTION OF THE VOLUME OF FLUID FIELD

The MCLS algorithm uses a simultaneously advanced VoF field to impose a correction on the LS field, to make the latter mass conserving. An evolution equation for the VoF field is derived by taken the derivative of (3), with respect to time:

$$\frac{d\Psi_k(t)}{dt} = \frac{d}{dt} \frac{1}{|\Omega_k|} \int_{\Omega_k} H(\Phi(\boldsymbol{x}, t)) d\Omega, \tag{42}$$

Because $H(\Phi(\boldsymbol{x}, t))$ is a material property and assuming a solenoidal velocity field $u_\alpha(\boldsymbol{x}, t)$, the right-hand side of (42) can be formulated as (dropping the explicit dependences on $\boldsymbol{x}$ and t)

$$\frac{1}{|\Omega_k|} \int_{\Omega_k} \frac{dH(\Phi)}{dt} d\Omega = -\frac{1}{|\Omega_k|} \int_{\Omega_k} H_{,\alpha}(\Phi) u_\alpha d\Omega = -\frac{1}{|\Omega_k|} \int_{\partial\Omega_k} H(\Phi) u_\alpha \hat{n}_\alpha dS, \tag{43}$$

where $\hat{n}_\alpha$ is the normal vector on the boundary of the element. Therefore, the evolution of the VoF field is governed by

$$\frac{d\Psi_k}{dt} = -\frac{1}{|\Omega_k|} \int_{\partial\Omega_k} H(\Phi) u_\alpha \hat{n}_\alpha dS \tag{44}$$

The fact that $H(\Phi)$ is C^{-1} continuous on $\partial\Omega$ makes accurate discretization of (44) that is very challenging. The application of 'standard' schemes developed for scalar hyperbolic equations leads either to strong oscillations in the solution or unacceptable smearing of the interface, jeopardizing mass conservation.

For Cartesian control volumes, accurate and efficient discretization schemes have been developed that utilize directional splitting, for example, the scheme presented in [10] and its strictly mass conserving adaptation [23]. Apart from the latter scheme, strict mass conservation can only be achieved with very intricate unsplit schemes.

In this paper, we propose to use a different approach for the advection of the VoF field. This approach is also used in [24] and is known as Eulerian–Lagrangian VoF evolution. The algorithm consists of three steps: Lagrangian advection of the color function, interface reconstruction, and remapping.

In the first step, each mesh element is considered as a material element and advected in a Lagrangian frame of reference. During this process the color function (a material property) is passively advected with the flow in each element. Because the velocity field is solenodial, each element retains its VoF value while being deformed and rotated. Furthermore, if a divergence free linear velocity field is assumed, this ensures a straight-sided triangle remains a straight-sided triangle, irrespective of how the relative position of the vertices changes (excluding folding). The assumption of a piecewise linear velocity distribution is consistent with the linear approximation of the interface.

The approach is based on using two grids. The first one is a (fixed) Eulerian grid and the second one a Lagrangian grid that consists of the same elements whose vertices have been translated. As the mesh elements are advected such that their area remains the same, this advection is consistent with Equation (43), that is, the advection is mass conserving.

The coordinates of the vertices of the Lagrangian grid are obtained by using a second order Runge–Kutta (RK) scheme, defined as

$$\boldsymbol{X}_{k_L}^{v\ n+\frac{1}{2}} = \boldsymbol{X}_{k_E}^{v} + \frac{\Delta t}{2}\mathbf{u}\left(\boldsymbol{X}_{k_E}^{v}, t^n\right), \tag{45}$$

$$\boldsymbol{X}_{k_L}^{n+1} = \boldsymbol{X}_{k_E}^{v} + \Delta t\mathbf{u}\left(\boldsymbol{X}_{k_L}^{n+\frac{1}{2}}, t^{n+\frac{1}{2}}\right), \tag{46}$$

where $k = 1.....,N_v$ and N_v is the total number of vertices. The $\boldsymbol{X}_{k_L}^{v\ n+1}$ represent the vertices of the Lagrangian mesh, $\boldsymbol{X}_{k_E}^{v}$ the vertices of the Eulerian mesh. Because of the assumptions on the velocity field, the advection defines an element-wise linear map from the Eulerian to the Lagrangian grid. The interface position in each element is defined by this mapping and does not need to be reconstructed. Once the interface position at the Lagrangian mesh is known, the color function can be mapped from the Lagrangian mesh back to the Eulerian mesh by means of a geometric process known as polygon-polygon *clipping* as described in [24–26].

In order to explain the clipping procedure, the elements of the Eulerian mesh are divided into two groups. We will refer to the cells that are completely filled with $Phase_0$ as *fully filled* cells. Those elements that contain both $Phase_0$ and $Phase_1$ will be referred to as *mixed cells*. In Figure 9 and 10, both cases are presented. A fully filled element is advected, as shown in Figure 9, using

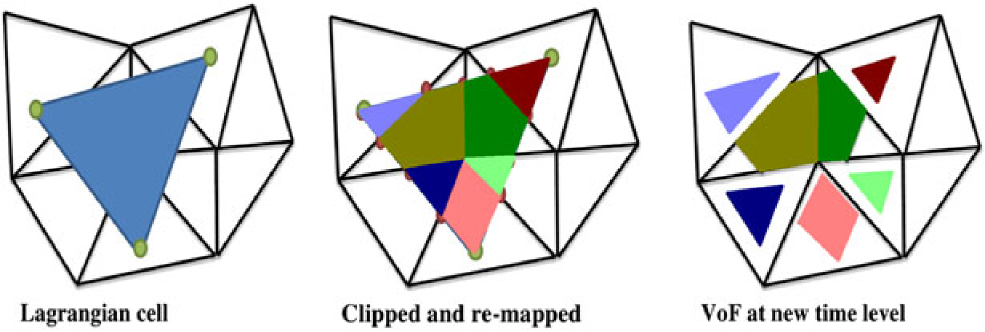


Figure 9. Advection, remapping, and distribution of the content of a control volume that is not intersected by the interface.

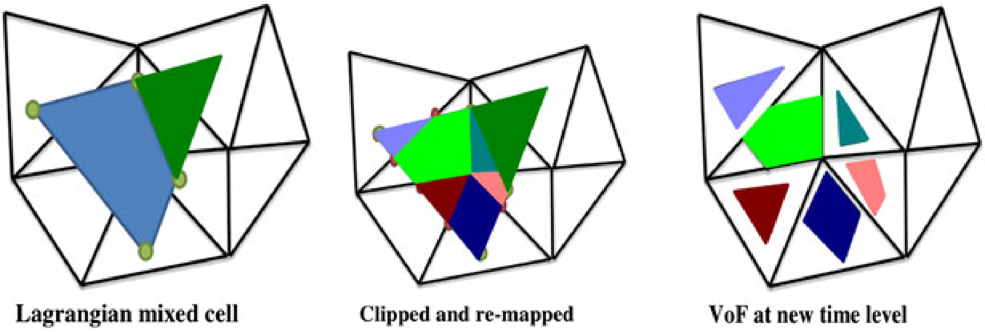


Figure 10. Advection, remapping and distribution of the content of a control volume that is intersected by the interface.

Copyright © 2015 John Wiley & Sons, Ltd.

Equations (45) and (46). This determines the position of the Lagrangian element corresponding to this Eulerian element. Now the fluid contained in this element needs to be distributed among the elements of the Eulerian grid. We refer to this process by *remapping*. Remapping of a fully filled cell on the Lagrangian grid is carried out by first locating the set of elements on the Eulerian grid that have a non-empty intersection with the Lagrangian element. Each of these intersections are polygonal regions defined by the coordinates of the intersections of the edges of the Lagrangian element with the Eulerian mesh.

A similar procedure is adopted for a mixed cell, as shown in Figure 10. In this case, the intersection of that part of the Lagrangian cell that is occupied by $Phase_0$ with the elements of the Eulerian mesh has to be determined. Each of these intersections are polygonal regions defined by the coordinates of the intersections of the edges of the Lagrangian element and the interface with the Eulerian mesh.

The area of the clipped region is computed by means of the following relation for the area of a closed polygon in 2D [27, 28]:

$$Area = \frac{1}{2}\left|\sum_{i=0}^{n-1}\left(x_1^i x_2^{i+1} - x_1^{i+1} x_2^i\right)\right|, \tag{47}$$

where i is the index for all the nodes and the first and last node coincide. In the current research, use is made of the MATLAB$^{\text{TM}}$ Mapping Toolbox to perform the polygon–polygon clipping. However, many different polygon clipping algorithms are available from image processing, for example, the Sutherland–Hodgeman algorithm and the Weiler–Atherton algorithm [29, 30].

6.1. *Mass conserving advection of the volume of fluid field for nonlinear velocity field*

The Eulerian–Lagrangian VoF advection method is accurately mass conserving when the velocity field is linear. In the case of a nonlinear velocity field, maintaining this accuracy presents a challenge. This is the result of the assumption that the triangular elements remain straight-sided triangles after advection. In reality, the nonlinear velocity field will deform the edges of the triangular element and representing it as a straight-sided triangle causes a loss or gain of mass (area) at the Lagrangian mesh. This deformation is shown in Figure 11. In the proposed algorithm, exact mass conservation is re-established using a method similar to the one proposed in (37). To prevent mass loss or gain due to the nonlinear velocity field, an intermediate step is introduced that is applied to the solution on the Lagrangian mesh before applying the clipping and redistribution algorithm. In this step, the total mass (area) at the Lagrangian mesh will be enforced to be equal to the total mass at the Eulerian mesh at previous time level t^n. This is performed by adjusting the volume fraction in all mixed

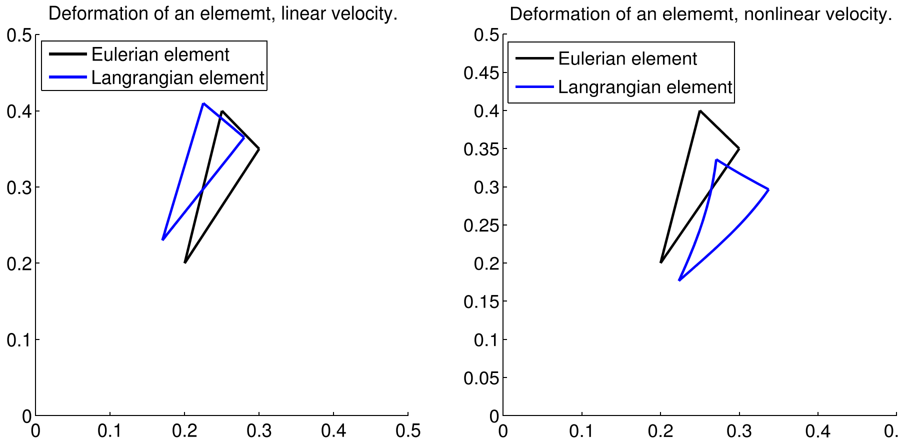


Figure 11. Straight-sided triangles remain straight-sided after being advected by a linear velocity field (left). When advected by a nonlinear velocity field this property is lost (right).

DOI: 10.1002/fld

cells such that the total mass error is compensated for. The complete procedure is explained in the following sections.

6.1.1. Mass error adjustment: Fully filled cells. Consider the fully filled k^{th} element of the Eulerian grid at time t^n. It has a volume fraction $\Psi_k^E = 1$ and area $\mid \Omega_k^E \mid$. Where superscript E refers to the Eulerian grid. The mass (M_k^E) enclosed by this element is equal to

$$M_k^E = \Psi_k^E \mid \Omega_k^E \mid . \tag{48}$$

Now, this element is advected with a nonlinear velocity field which results in a deformed triangle so the area enclosed by the element is not the same as on the Eulerian grid, that is, a mass error has incurred. However, during advection, the fully filled cell should remain fully filled, so it has volume fraction $\Psi_k^L = 1$ and the area $\mid \Omega_k^L \mid$. Where superscript L represents the Lagrangian grid. The mass (M_k^L) enclosed by the Lagrangian element is

$$M_k^L = \Psi_k^L \mid \Omega_k^L \mid . \tag{49}$$

The mass lost or gained by the element due to the deformation caused by the nonlinear velocity field is

$$M_k^{err.} = M_k^E - M_k^L, \tag{50}$$

where $M_k^{err.}$ is the mass error of the k^{th} fully filled cell. If there are N_f fully filled elements that are advected, using a nonlinear velocity field, then the total mass error $(TM^{err.})$ caused by the fully filled cells can be defined as

$$TM^{err.} = \sum_{i=1}^{N_f} \left(M_i^E - M_i^L \right). \tag{51}$$

The total mass error computed in Equation (51) is distributed evenly among the mixed cells. If there are N_m mixed cells then the average mass error (M_{avg}) can be defined as

$$M_{avg.} = \sum_{i=1}^{N_f} \left(\frac{M_i^E - M_i^L}{N_m} \right). \tag{52}$$

This average mass error is added to each of the mixed cells to ensure global mass conservation.

6.1.2. Mass error adjustment: partially filled cells. In case of a partially filled or mixed cell, a different procedure is adopted. The interface position is adjusted in each mixed control volume in the Lagrangian grid such that the mass contained within is equal to the mass contained within the corresponding control volume on the Eulerian grid plus the average mass error that follows from (52). Both the position and the orientation of the interface have to be determined. The orientation of the interface can be determined from the LS field, but this approach would require the interpolation of the LS field from the Eulerian to the Lagrangian grid at time t^{n+1}. This idea is not computationally efficient; for details, see [24]. To mitigate this problem, a different procedure is used, which is based on the assumption that the interface can be reconstructed on the Lagrangian element by advecting the two points that represent the interface in the corresponding Eulerian element. Naturally, this puts a threshold on the size of the time step that can be taken. However, the idea is computationally efficient and also in line with the idea of distributing the average mass error to the mixed cells. The details of the interface adjustment for the control volumes in the Lagrangian mesh are presented in the following subsection.

DOI: 10.1002/fld

6.1.3. Interface adjustment for partially filled cells. Consider a partially filled cell at the Eulerian grid at time t^n, as shown in Figure 12(a). The vertices of the element are $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_1}}, \boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_2}}$, and $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_3}}$. The interface is represented by a linear polynomial, that is, a line segment defined by its end points $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_{31}}}$ and $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_{32}}}$. Also, the area of the triangle formed by $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_3}}, \boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_{31}}}$, and $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_{32}}}$ represents the exact amount of fluid of interest in this element. The location of the points $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_{31}}}$ and $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_{32}}}$ is obtained by using the LS information of the element. Assume vertex $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_3}}$ is in the negative LS region, then $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_1}}$ and $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_2}}$ are in the positive region and the interface normal points toward the positive LS region. Using the linear interpolation parameters β_i for $i = 1, 2$, the intersection of the interface with the edges $\overline{\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_3}} \boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_1}}}$ and $\overline{\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_3}} \boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_2}}}$ can be found in the following way:

$$\beta_i = \frac{-\Phi\left(\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_3}}\right)}{\Phi\left(\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_3}}\right) - \Phi\left(\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_i}}\right)}, \tag{53}$$

where $\Phi\left(\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_1}}\right), \Phi\left(\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_2}}\right)$ and $\Phi\left(\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_3}}\right)$ are the LS values at the vertices $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_1}}, \boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_2}}$, and $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_3}}$, respectively. Using the parameters β_i, the interface end points $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_{31}}}$ and $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_{32}}}$ can be computed as

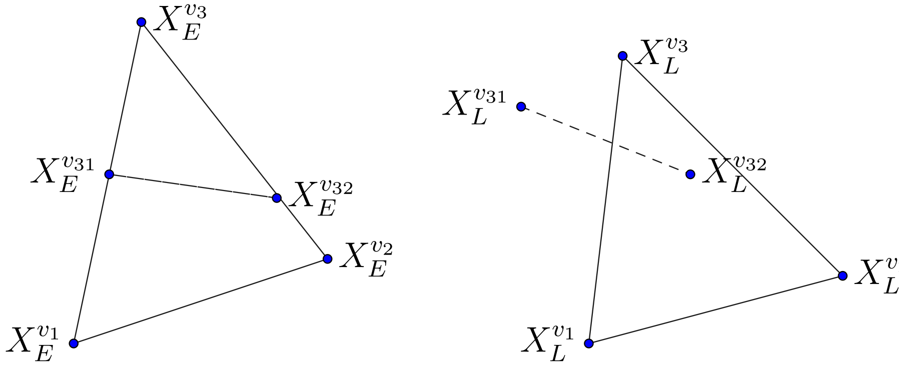
$$\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_{3(i)}}} = \beta_i \boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v(i)}} + (1 - \beta_i)\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_3}} \quad i = 1, 2. \tag{54}$$

Once the extremities of the interface are known, they are advected along with the vertices of the element, using Equations (45) and (46). This defines the corresponding Lagrangian element as shown in Figure 12(b). The corresponding vertices on the Lagrangian elements are $\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v_1}}, \boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v_2}}$, and $\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v_3}}$, and the interface extremities are $\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v_{31}}}$ and $\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v_{32}}}$, respectively.

The end points of the interface are not necessarily located on the edges. This is because the triangle has deformed. In order to avoid overlapping with neighboring elements, the interface end points are projected on the edges of the element. This is performed by computing the intersection of the edges $\overline{\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v_3}} \boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v_1}}}$ and $\overline{\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v_3}} \boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v_2}}}$ with the interface. This defines points $\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v_{31}}'}$ and $\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v_{32}}'}$, as shown in Figure 13(a). The interface orientation is still the same, the only difference is that it is mapped to the edges of the Lagrangian element. Clearly, the interface can be reconstructed in a straightforward way on the Lagrangian mesh without the need for information of the LS field.

After the interface reconstruction step, the goal is to achieve exact mass conservation, that is, the interface has to be shifted from its current position to make sure the mass within the Lagrangian control volume corresponds to the mass in the corresponding Eulerian control volume plus the average mass error defined by (52).

Assume the area of the Eulerian element is $A_E = \blacktriangle \boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_3}} \boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_{31}}} \boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_{32}}}$, and the area of the Lagrangian element is $A_L = \blacktriangle \boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v_3}} \boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v_{31}}'} \boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v_{32}}'}$. Now the goal is to find the points on the edges



(a) Eulerian element. (b) Lagrangian element.

Figure 12. The position of the interface in a mixed cell in the Lagrangian mesh is based on the advection of the extremities of the interface in the corresponding mixed cell in the Eulerian mesh.

DOI: 10.1002/fld

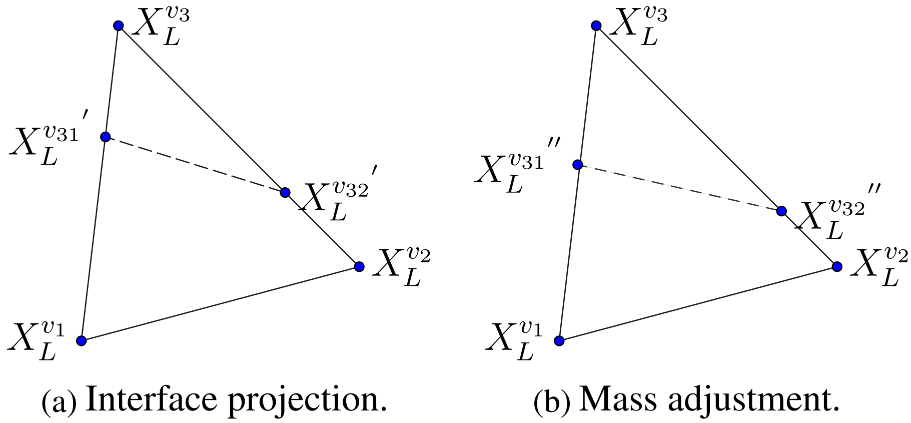


Figure 13. For a nonlinear velocity field, the advected extremities of the interface are first projected on the edges of the Lagrangian control volume, before the appropriate interface position can be determined.

$\overline{\boldsymbol{X_L^{v_3} X_L^{v_1}}}$ and $\overline{\boldsymbol{X_L^{v_3} X_L^{v_2}}}$, such that the area enclosed by these points and the vertex $\boldsymbol{X_L^{v_3}}$ is equal to A_E plus $M_{avg.}$. This is achieved by computing the following ratio:

$$\theta = \sqrt{\frac{A_E + M_{avg.}}{A_L}}. \tag{55}$$

Using, the value θ points on the edges can be computed as follows

$$\boldsymbol{X_L^{v_{3(i)''}}} = \theta \boldsymbol{X_L^{v_{3(i)'}}} + (1-\theta)\boldsymbol{X_L^{v_3}}, \tag{56}$$

where the points $\boldsymbol{X_L^{v_{3(i)''}}}$, for $i = 1, 2$, correspond to the edges $\overline{\boldsymbol{X_L^{v_3} X_L^{v_1}}}$ and $\overline{\boldsymbol{X_L^{v_3} X_L^{v_2}}}$, respectively, as shown in Figure 13(b). This makes sure that the area is identical to the area of the Eulerian cell. This holds for the case where the fluid of interest is within a triangular region $\boldsymbol{X_L^{v_3} X_L^{v_{31'}} X_L^{v_{32'}}}$. For the region enclosed by the points $\boldsymbol{X_L^{v_1} X_L^{v_2} X_L^{v_{32'}} X_L^{v_{31'}}}$, a slightly different procedure is adopted. Here, it is important to note that as the triangles are allowed to deform during advection, the total area of the Lagrangian element is not equal to the total area of the corresponding Eulerian element. Therefore, using Equation (55), to conserve the area of the region by subtracting the area enclosed in $\boldsymbol{X_L^{v_1} X_L^{v_2} X_L^{v_{32'}} X_L^{v_{31'}}}$ from A_E will not conserve the region $\boldsymbol{X_L^{v_1} X_L^{v_2} X_L^{v_{32'}} X_L^{v_{31'}}}$ on the Lagrangian mesh. In order to circumvent this problem, Equation (55) is modified. Figure 13 is used again to explain the procedure, as all other steps except the use of Equation (55) are the same. Consider a fluid of interest enclosed in the region $A_E = \blacksquare \boldsymbol{X_E^{v_1} X_E^{v_2} X_E^{v_{32}} X_E^{v_{31}}}$ at the Eulerian mesh. After advection and projecting the interface points at the edges of the Lagrangian cell, the corresponding Lagrangian element is $A_L = \blacksquare \boldsymbol{X_L^{v_1} X_L^{v_2} X_L^{v_{32'}} X_L^{v_{31'}}}$ and the total area of the Lagrangian element is $A_{LT} = \blacktriangle \boldsymbol{X_L^{v_1} X_L^{v_2} X_L^{v_3}}$. Using this information, the modified form of Equation (55) is obtained as

$$\theta = \sqrt{\frac{A_{LT} - (A_E + M_{avg.})}{A_{LT} - A_L}}. \tag{57}$$

Similarly, points $\boldsymbol{X_L^{v_{3(i)''}}}$ for $i = 1, 2$, on the edges are computed, using (56) . This will ensure the mass (area) of the cell corresponds to the mass of the Eulerian cell. Once mass conservation is achieved at the Lagrangian mesh, the clipping procedure is applied to map the VoF to the Eulerian mesh for time t^{n+1}.

7. INVERSE FUNCTION FOR THE LEVEL-SET CORRECTION

To make the LS mass-conserving, a correction is applied based on the advected VoF field. In this approach, the advected LS field Φ_k^* is used to compute a tentative VoF field $\Psi^* = f\left(\Phi_k^*, \nabla\Phi_k^*\right)$ in every mixed cell, using the VoF function (32). The tentative VoF field is compared locally with the Eulerian–Lagrangian advected VoF field Ψ_k^{n+1}. If the difference is within a given tolerance then there is no need to correct the LS in that particular cell, as it is already mass-conserving, that is, the LS field complies with (3), and $\Phi_k^{n+1} = \Phi_k^*$. However, if this difference exceeds the prescribed tolerance, a correction is required. In order to perform the correction, the inverse of $f(\Phi_k, \nabla\Phi_k)$ is required. This function is referred to as the *inverse VoF function* $g(\Psi_k, \nabla\Phi_k)$. This function is derived from (32).

Assume the value of the VoF field for Ω_k is given by

$$\Psi_k^{n+1} = f\left(\Phi_k^{n+1}, \nabla\Phi_k^{n+1}\right), \tag{58}$$

for yet unknown Φ_k^{n+1}. Because the mass conserving corrections that need to be imposed in the MCLS are very small, the orientation of the interface is assumed to remain unchanged. Therefore, we can replace (58) by

$$\Psi_k^{n+1} = f\left(\Phi_k^{n+1}, \nabla\Phi_k^*\right). \tag{59}$$

This means that in the conversion, the height of the interface is adjusted without changing its orientation. For a triangular (or more generally a convex) element, there is a unique value of Φ_k^{n+1} that fulfills (59) for $\Psi_k \in [0, 1]$. Therefore, the inverse of $f(\Phi_k, \nabla\Phi_k)$ with respect to its first argument can be defined as

$$\Phi_k = g(\Psi_k, \nabla\Phi_k). \tag{60}$$

In order to obtain a uniformly valid inverse function, a switching parameter similar to (31) can be defined as

$$S_p'(\Psi_k) = \begin{cases} 1 & : \Psi_k \geqslant \dfrac{D_{\xi_2}}{D_{\xi_1}} \\ 0 & : \Psi_k < \dfrac{D_{\xi_2}}{D_{\xi_1}} \end{cases}. \tag{61}$$

A critical value of the VoF is $\Psi_k = \dfrac{D_{\xi_2}}{D_{\xi_1}}$, because for this value the interface is passing through the vertex $\boldsymbol{\xi}^A$; see Figure 7. Using (61), (27) can be written as

$$\Psi_k = \frac{(\tilde{\phi}(\mathbf{0}))^2}{D_{\xi_1} D_{\xi_2}} - S_p'(\Psi_k)\frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})^2}{D_{\xi_2}(D_{\xi_1} - D_{\xi_2})}. \tag{62}$$

Define the coefficients $c_{1,2}'$ as

$$c_1' = \frac{1}{D_{\xi_1} D_{\xi_2}}, \qquad c_2' = \frac{-S_p'(\Psi_k)}{D_{\xi_2}(D_{\xi_1} - D_{\xi_2})}. \tag{63}$$

Now $\tilde{\phi}(\mathbf{0})$ is the unique root of the following quadratic equation:

$$\begin{aligned} &\left(c_1' + c_2'\right)(\tilde{\phi}^{n+1}(\mathbf{0}))^2 + 2c_2' D_{\xi_2}(\tilde{\phi}^{n+1}(\mathbf{0})) + c_2' D_{\xi_2}^2 - \Psi_k = 0, \\ &\forall\ \Psi_k \in [0, 1]. \end{aligned} \tag{64}$$

within the interval $\left[-D_{\xi_1}, 0\right]$. This unique root is given by

$$\tilde{\phi}(\mathbf{0}) = -\left(2c_2' D_{\xi_2}\right) - \frac{\sqrt{\left(2c_2' D_{\xi_2}\right)^2 - 4\left(c_1' + c_2'\right)\left(c_2' D_{\xi_2}^2 - \Psi_k\right)}}{2\left(c_1' + c_2'\right)}. \tag{65}$$

The inverse function $g(\Phi_k, \nabla\Phi_k)$ can now be defined as

$$\begin{aligned} \Phi_k &= g(\Psi_k, \nabla\Phi_k) : [0,1] \to \left[\frac{-2D_{\xi_1} + D_{\xi_2}}{3}, \frac{D_{\xi_1} + D_{\xi_2}}{3}\right], \\ g(\Psi_k, \nabla\Phi_k) &= -\left(2c_2' D_{\xi_2}\right) - \frac{\sqrt{\left(2c_2' D_{\xi_2}\right)^2 - 4\left(c_1' + c_2'\right)\left(c_2' D_{\xi_2}^2 - \Psi_k\right)}}{2\left(c_1' + c_2'\right)}) + \frac{D_{\xi_1} + D_{\xi_2}}{3}, \end{aligned} \tag{66}$$

where $[D_{\xi_1} \quad D_{\xi_2}]^T = J\nabla\Phi_k$. The conversion of LS to VoF depends on the value and gradient of the LS at the center of the control volume. The key difference between the original MCLS and the proposed algorithm is that in the latter, the discretization of the LS field with a discontinuous Galerkin method leads to a local, element-wise definition of the LS field. This means that the LS field can be corrected locally, contrary to the original MCLS algorithm where an iterative procedure is used to have both Φ_k^{n+1} and $\nabla\Phi_k^{n+1}$ comply with (58). This is a major improvement with respect to efficiency and robustness in comparison with the original MCLS algorithm.

8. COMPUTATIONAL COST OF THE MASS-CONSERVING CORRECTION

Accurately, mass-conserving advection of the VoF field on a discretization of general unstructured control volumes is very challenging but necessary in many but certainly not all applications. Incorporating this correction makes the algorithm significantly more costly than a 'pure' LS approach, but comparable in cost to algorithms that are based on an explicit reconstruction of the interface. The Eulerian–Lagrangian approach of advection and the required 'clipping' algorithm are computationally intensive. However, these 'clipping' type of algorithms are used extensively in computer graphics and under continuous development. Furthermore, they are highly parallelizable and readily portable to GPU architectures. Alternative algorithms, for example, based on the use of compressive schemes used for modified LS formulations are currently investigated. The use of a discontinuous Galerkin discretization and the resulting locality of the mass conserving correction lead to a significant gain in efficiency with respect to the original formulation of the MCLS method.

9. TEST CASES

The key characteristics of the method that need to be validated are the mass conservation property and the accuracy of the LS field. Therefore, three test cases are selected based on different velocity fields: constant, linear, and nonlinear. The test cases chosen are the translation of a circular interface with a constant velocity, the rotation of a circular interface around the center of the domain with linear velocity and the very popular single reverse vortex test case [10, 24]. These test cases are sufficient to demonstrate both the mass conservation properties and accuracy of the method. Apart from these test cases, the accuracy of the conversion between the LS field and the VoF field is considered separately using the VoF function and its inverse. Also the initial mass error that is incurred due to the piecewise linear representation of the interface is analyzed. This test case is referred to as interface back and forth reconstruction. We define the mass error E_M^n and the discrete L_2 error in the LS field $E_\Psi(t^n)$ as

DOI: 10.1002/fld

$$
\begin{aligned}
E_M^n &= M^n - M_{\text{exact}}(t^n),\, M^n = \sum_{k=1}^{K} \Psi_k^n |\Omega_k|, \\
E_\Phi(t^n) &= \sqrt{\frac{\sum_{k=1}^{K} \left(\Phi_K^n - \Phi\left(\boldsymbol{X}_k, t^n\right)_{\text{exact}}\right)^2}{K}},
\end{aligned}
\tag{67}
$$

where $M_{\text{exact}}(t^n)$ and $\Phi\left(\boldsymbol{X}_k, t^n\right)_{\text{exact}}$ are the exact area contained within the interface and the exact LS field, respectively. The order of $E_\Phi(t^n)$ is estimated through Richardson extrapolation, on a sequence of unstructured meshes. We expect to find second order convergence of both $E_\Phi(t^n)$ and E_M^n because of the use of linear basis functions for the LS field and the use of the linearized LS field in the conversion to VoF. The VoF advection scheme is mass conserving to machine precision, so apart from an initialization error, the mass loss should be virtually negligible over the duration of the simulations. We will only focus on the spatial accuracy of the discretization. In each test case, all quantities are made dimensionless by introducing reference length and time scales equal to 1 m and 1 s, respectively.

9.1. Conversion between level-set and volume of fluid representation

This test case is considered to demonstrate the accuracy of the conversion of the LS field to the VoF field and vice versa using (35) and its inverse (66): the VoF field is converted to the LS field using Equation (64), without involving any advection of the interface. Naturally, interconversion of the LS field and the VoF field without loss of mass is a prerequisite for the algorithm to achieve (nearly) exact mass conservation. This test shows that the error induced by the interconversion is negligible. A circular fluid region is considered with radius 0.2 in a domain of size $\Omega_D = [0,1] \times [0,1]$. The initial condition for the LS field is given as

$$
\Phi(\boldsymbol{x}, 0) = \tfrac{1}{25} - \left(x_1 - \tfrac{1}{2}\right)^2 - \left(x_2 - \tfrac{1}{2}\right)^2, \quad \boldsymbol{x} \in \Omega_D. \tag{68}
$$

The corresponding VoF field is given by

$$
\Psi(\boldsymbol{x}, 0) = \begin{cases} 1, & \left(x_1 - \tfrac{1}{2}\right)^2 + \left(x_2 - \tfrac{1}{2}\right)^2 - \tfrac{1}{25} < 0, \\ 0, & \left(x_1 - \tfrac{1}{2}\right)^2 + \left(x_2 - \tfrac{1}{2}\right)^2 - \tfrac{1}{25} > 0. \end{cases}, \quad \boldsymbol{x} \in \Omega_D. \tag{69}
$$

Furthermore, if we consider $Phase_1$ to be the fluid of interest, the total area of the domain of interest is $M_{\text{exact}} = \frac{\pi}{25}$. The interface is represented by the zero-contour of the LS field and the interface normal points outwards from the negative LS region. Firstly, the LS field is converted into the VoF field using (32). Secondly, the obtained VoF field is converted back to the LS field using (66). After that the difference between the LS field before and after the conversion is evaluated for all mixed cells. We found that the LS field remains unchanged up to machine precision.

In the MCLS, the interface is approximated by a set of line segments, because of the fact that the conversion between LS and VoF is based on the linearized LS field. Depending on the procedure for the initialization, this approximation will incur an error in the mass(area) when the LS field is initialized exactly, or an error in the LS field when the VoF field is initialized exactly. We consider (68) as initial condition for the LS field. In Table I, E_M is presented for four different characteristic

Table I. Accuracy analysis of the interface back-and-forth conversion test case. Initialization with the exact level-set field introduces a mass error that converges with second order accuracy.

Characteristic meshwidth h	M^n	$\lvert E_M \rvert$	Order
3.64e-02	1.2456e-01	1.10008e-03	—
2.43e-02	1.2515e-01	5.0951e-04	2.16
1.67e-02	1.2540e-01	2.5679e-04	1.98
1.02e-02	1.2556e-01	9.4813e-05	2.71

DOI: 10.1002/fld

mesh widths $h = \sqrt{\Sigma|\Omega_k|/N}$. Clearly, $E_M = \mathcal{O}(h^2)$, and the initial error strongly dominates the error incurred by the back-and-forth conversion. Alternatively, the LS field can be initialized by conversion of the *exact VoF field* (69). This approach will incur an error in the LS field as shown in Figure 14, where the zero level countour of the LS field initialized with (68) is compared with the LS field converted from (69). Because the conversion between the two fields is based on the linearized LS field, the conversion induces a small movement of the interface, which can be shown to decrease with second order upon grid refinement. Clearly, to achieve nearly exact mass conservation, it is essential to initialize the LS field by converting the initial condition for the VoF field.

9.2. *Translation of circular region*

In this test case, the translation of a circular interface of radius 0.15 is considered in a domain of size $\Omega_D = [0,1] \times [0,1]$. Initially, the interface $\phi(\boldsymbol{x},0) = 0$ is centered at $(0.5, 0.2)^T$. The LS and VoF fields are advected with a constant velocity field $\boldsymbol{u}(\boldsymbol{x}) = (0, 0.1)^T$ for $0 < t \leqslant T = 5$. The exact LS field at the final time is a circular region with unchanged radius but centered at $(0.5, 0.7)^T$. In Table II, the average mass error $\overline{E_M^n}$ and the error of the LS field $E_\Phi(T)$ are presented for three different mesh sizes. The results indicate that the solution of the LS field converges with second order accuracy.

9.3. *Rotation of circular region around the center of the domain*

The third test case concerns the rotation of an initially circular interface. Consider a circular region of radius 0.15 in a domain of size $\Omega_D = [0,1] \times [0,1]$. Initially, the interface $\phi(\boldsymbol{x},0) = 0$ is centered around $(0.5, 0.75)^T$. The LS and VoF fields are advected with a divergence free linear velocity field $\boldsymbol{u}(\boldsymbol{x})$ defined as

$$\boldsymbol{u}(\boldsymbol{x}) = (x_1 - 0.5, -x_2 + 0.5)^T, \ \boldsymbol{x} \in \Omega_D. \tag{70}$$

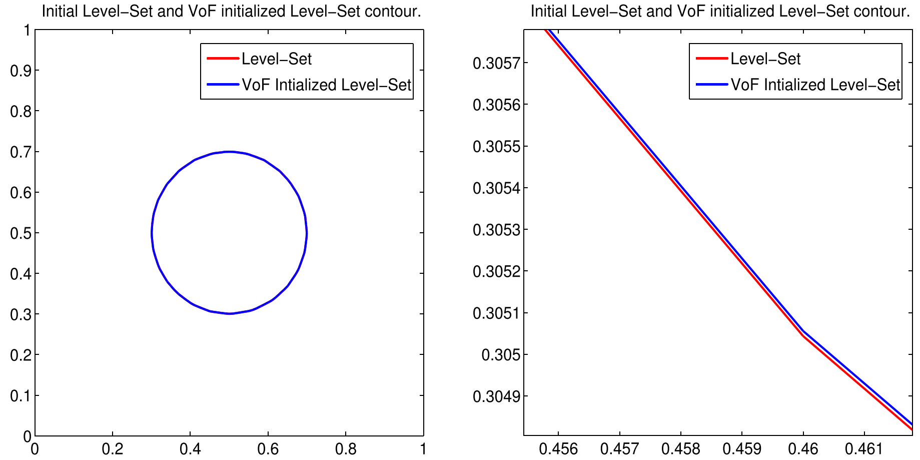


Figure 14. Initialization with the exact volume-of-fluid field leads to a slightly different initial position of the interface but eliminates the mass error incurred by initialization with the exact level-set field.

Table II. Average mass error $\overline{E_M^n}$ and error in the level-set field $E_\Phi(T)$ for the translation of a circular interface.

Characteristic meshwidth h	$\overline{E_M^n}$	$E_\Phi(T)$	Order of $E_\Psi(T)$
3.64e-02	2.6835E-15	8.0531E-04	—
2.43e-02	2.2830E-15	5.1876E-04	1.55
1.67e-02	8.2976E-16	2.3053E-04	2.25

DOI: 10.1002/fld

for $0 < t \leqslant T = 2\pi$. At $t = T$ the LS field is compared with the initial LS field as the interface should return to its initial position after one complete revolution.

Figure 15 shows the time history of the mass error during one full rotation for three different mesh sizes. In Table III, the average mass error $\overline{E_M^n}$ and the error of the LS field $E_\Phi(T)$ are presented. The solution of the LS field converges with second order accuracy.

9.4. *The reverse vortex test case*

The reverse vortex or single reverse vortex test case is one of the more challenging tests for models for immiscible two-phase flow. The reason for this is the severe stretching and deformation of the convected interface caused by the nonlinear velocity field during the advection which makes accurate mass conservation challenging.

Consider an initially circular fluid region of radius 0.15 in a domain of size $\Omega_D = [0,1] \times [0,1]$. Initially, the center of the fluid is at $(0.5, 0.75)$. This circular region is advected with a divergence free nonlinear velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$ defined as

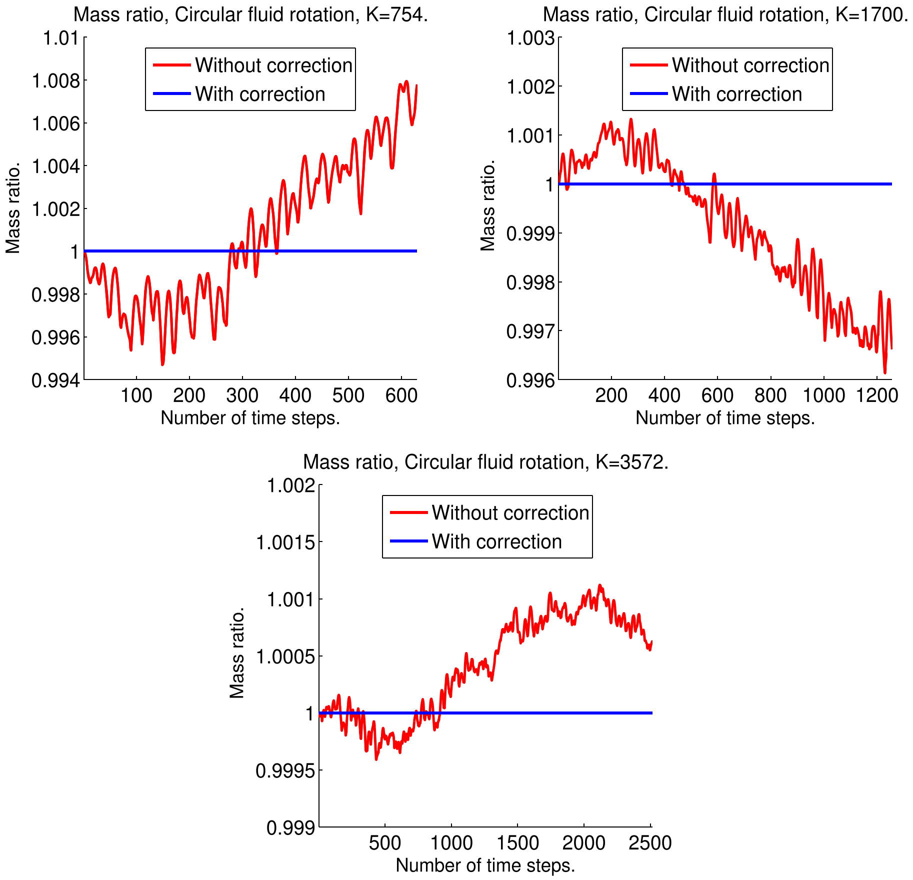


Figure 15. Time history of mass error for the circular fluid rotation case.

Table III. Average mass error $\overline{E_M^n}$ and error in the level-set field $E_\Phi(T)$ for the rotation of a circular interface.

Characteristic mesh width h	$\overline{E_M^n}$	$E_\Psi(T)$	Order of $E_\Psi(T)$
3.64e-02	1.6199E-14	6.6142E-03	—
2.43e-02	4.2944E-16	2.2104E-03	2.99
1.67e-02	1.7417E-15	1.2526E-03	1.77

DOI: 10.1002/fld

$$\boldsymbol{u}(\boldsymbol{x},t) = (\sin^2(\pi x_1)\sin(2\pi x_2), -\sin^2(\pi x_2)\sin(2\pi x_1))^T \cos\left(\frac{\pi t}{T}\right), \quad \boldsymbol{x} \in \Omega_D,\ 0 \leqslant t \leqslant T. \tag{71}$$

This definition of the velocity field implies that the interface should return to its original position at time $t = T$, which is also the final time. The value of T is set to 2.

The evolution of the interface is shown in Figure 16 for three different mesh widths, together with the exact initial and final position of the interface.

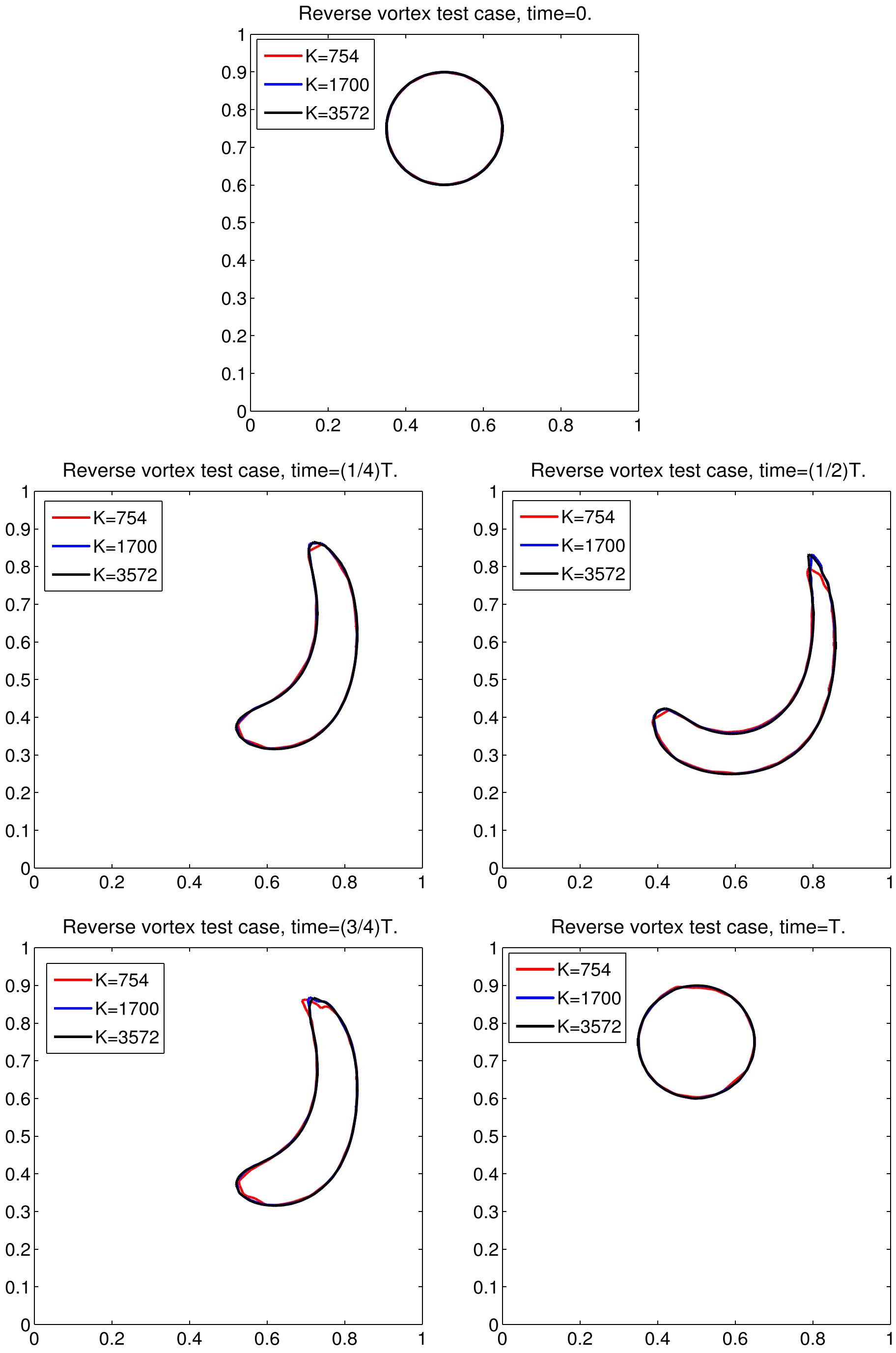


Figure 16. The evolution of the interface position for the reverse vortex test case.

DOI: 10.1002/fld

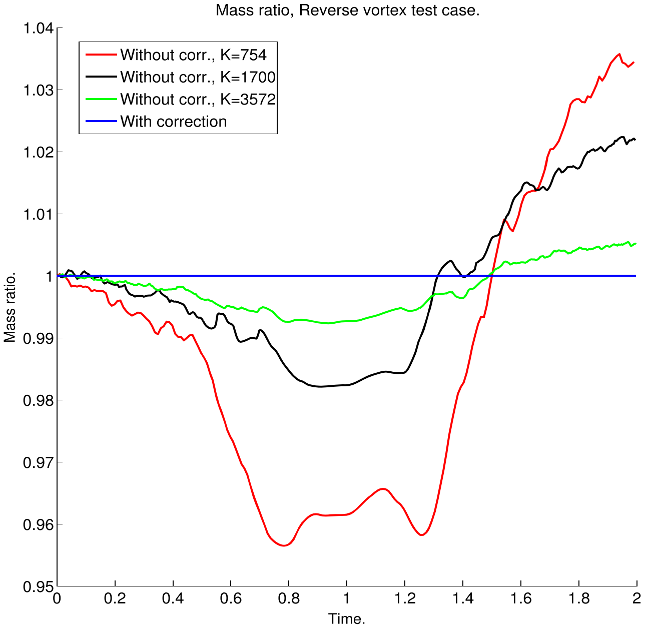


Figure 17. Time history of mass error for the reverse vortex test case.

Table IV. Average mass error $\overline{E_M^n}$ and error in the level-set field $E_\Phi(T)$ for the reverse vortex test case.

Characteristic mesh width h	$\overline{E_M^n}$	$E_\Psi(T)$	Order of $E_\Psi(T)$
3.64e-02	1.48167E-14	5.6716E-3	—
2.43e-02	3.3508E-16	1.8284E-3	3.101
1.67e-02	4.7809E-16	1.0951E-3	1.669

Figure 17 shows the time history of the mass error during one full rotation for three different mesh sizes. In Table IV, the average mass error $\overline{E_M^n}$ and the error of the LS field $E_\Phi(T)$ are presented. The solution of the LS field converges with second order accuracy, while even for this nonlinear, velocity field mass is conserved (nearly) exactly.

10. CONCLUSIONS

We have developed a (nearly) exactly mass-conserving method for the simulation of immiscible incompressible two phase flow in geometrically intricate two-dimensional domains. The method developed is the extension of the MCLS method toward unstructured triangular grids. The VoF function and the inverse function derived for a triangular mesh are very simple, robust, and efficient to evaluate. Due to the use of a discontinuous Galerkin discretization method for the LS field, the mass-conserving correction can be applied locally in each element. This approach is significantly more efficient and robust than the original MCLS formulation. Numerical experiments indicate the LS field converges with second order accuracy in space and mass is conserved up to machine precision.

REFERENCES

1. Rider W, Kothe D. A marker particle method for interface tracking. *Proceedings of the Sixth International Symposium on Computational Fluid Dynamics*, Lake Tahoe, Nevada, 1995; 976–981.
2. Hirt CW, Nichols BD. Volume of fluid VoF method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**(1):201–225.

3. Osher S, Sethian JA. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics* 1988; **79**(1):12–49.
4. Tryggvason G, Bunner B, Esmaeeli A, Juric D, Al-Rawahi N, Tauber W, Han J, Nas S, Jan YJ. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics* 2001; **169**(2):708–759.
5. Huang M, Chen B, Wu L. A {SLIC-VOF} method based on unstructured grid. *Microgravity Science and Technology* 2010; **22**(3):305–314.
6. Lopez J, Hernandez J, Gomez P, Faura F. An improved PLIC-VOF method for tracking thin fluid structures in incompressible two-phase flows. *Journal of Computational Physics* 2005; **208**(1):51–74.
7. Kimmel R. The Osher–Sethian level set method. In *Numerical Geometry of Images*. Springer: New York, 2004; 50–60.
8. Sussman M. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics* 2003; **187**(1):110–136.
9. Wang Y, Simakhina S, Sussman M. A hybrid level set-volume constraint method for incompressible two-phase flow. *Journal of Computational Physics* 2012; **231**(19):6438–6471.
10. Sussman M, Puckett EG. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics* 2000; **162**(2):301–337.
11. van der Pijl SP, Segal A, Vuik C, Wesseling P. Computing three-dimensional two-phase flows with a mass-conserving level-set method. *Computing and Visualization in Science* 2008; **11**:221–235.
12. Yang X, James AJ. Analytic relations for reconstructing piecewise linear interfaces in triangular and tetrahedral grids. *Journal of Computational Physics* 2006; **214**(1):41–54.
13. Lv X, Zou Q, Zhao Y, Reeve D. A novel coupled level set and volume of fluid method for sharp interface capturing on 3D tetrahedral grids. *Journal of Computational Physics* 2010; **229**(7):2573–2604.
14. Olsson E, Kreiss G. A conservative level set method for two phase flow. *Journal of Computational Physics* 2005; **210**(1):225–246.
15. Olsson E, Kreiss G, Zahedi S. A conservative level set method for two phase flow {II}. *Journal of Computational Physics* 2007; **225**(1):785–807.
16. Desjardins O, Moureau V, Pitsch H. An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *Journal of Computational Physics* 2008; **227**(18):8395–8416.
17. wang Shu C. Essentially Non-oscillatory and Weighted Essentially Non-oscillatory Schemes for Hyperbolic Conservation Laws. In *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, vol. 1697, Lecture Notes in Mathematics. Springer-Verlag: Berlin Heidelberg, 1998; 325–432.
18. Jiang GS, Shu CW. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics* 1996; **126**(1):202–228.
19. Giraldo FX, Hesthaven JS, Warburton T. Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations. *Journal of Computational Physics* 2002; **181**(2):499–525.
20. Hesthaven JS, Warburton T. Discontinuous Galerkin methods for the time-domain Maxwell's equations. *ACES Newsletter* 2004; **19**:10–29.
21. Grooss J, Hesthaven JS. A level set discontinuous Galerkin method for free surface flows. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**(25-28):3406–3429.
22. Marchandise E, Remacle JF, Chevaugeon N. A quadrature-free discontinuous Galerkin method for the level set equation. *Journal of Computational Physics* 2006; **212**(1):338–357.
23. Weymouth GD, Yue DKP. Conservative volume-of-fluid method for free-surface simulations on Cartesian-grids. *Journal of Computational Physics* 2010; **229**(8):2853–2865.
24. Yang X, James AJ, Lowengrub J, Zheng X, Cristini V. An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids. *Journal of Computational Physics* 2006; **217**(2):364–394.
25. Zheng X, Lowengrub J, Anderson A, Cristini V. Adaptive unstructured volume remeshing-II: application to two- and three-dimensional level-set simulations of multiphase flow. *Journal of Computational Physics* 2005; **208**(2): 626–650.
26. Anderson A, Zheng X, Cristini V. Adaptive unstructured volume remeshing-I: the method. *Journal of Computational Physics* 2005; **208**(2):616–625.
27. Sunday D. Fast polygon area and newell normal computation. *Journal of Graphics Tools* 2002; **7**(2):9–13.
28. Gelder AV. Efficient computation of polygon area and polyhedron volume. In *Graphics Gems V*, Paeth AW (ed.). Academic Press: San Diego, CA, 1995; 35–41.
29. Révész T. Clipping polygons with sutherland-hodgman's algorithm. *C Users Journal* 1993; **11**(8):23–34.
30. Chakraborty A. *An Extension of Weiler–Atherton Algorithm to Cope with the Self-Intersecting Polygon.* CoRR, vol. abs/1403.0917. Cornell University Library: Ithaca, US, 2014.

DOI: 10.1002/fld