A coupled level set and volume of fluid method for sharp interface simulation

Linfan Zhang

March 27, 2019

Abstract

This paper presents a coupled level-set and volume-of-fluid method for unstructured meshes. This method designed for simulating incompressible two phase flows combines both the advantages of LS method and VOF method. The method is called CLSAdvector, and is implemented into OpenFOAM® as open source. Volume of fluid (VOF) idea is conservative because it can calculate the volume of one of the fluids transported across the mesh faces in a time step. In contrast to VOF, the LS method provides a sharp interface and a smooth transition in the physical properties across the interface. The novelty of the CLSAdvector concept combines the ideas of VOF and LS method. First, an algorithm is designed for calculating the position of interface inside cells where the void fraction and the direction of interface are given. Second, the level set function is limited in a narrow band that contains the interface, which improves the efficiency of the algorithm. The feasibility and accuracy of the current method are validated by several cases including Zalesak's problem, 2D vortex, dam break.

1 Introduction

Incompressible two phase flow with large density ratio at the free surface can be found in many natural phenomenon and industrial processes such as power plant, internal combustion, chemical reactor. Especially during nuclear reactor severe accidents, the molten corium from the fuel core region or the reactor pressure vessel (RPV) may fall into water pool. Then fuel coolant interaction (FCI) will happen and can lead to steam explosion which can cause severe damage to the containment of the nuclear reactor. Severity of steam explosion can be decided by the limited breakup of the molten pour stream flowing through the water prior to stream explosion[1]. In order to study the effect of melt physical properties on the jet breakup length and the characterization of droplet size distribution in the premixing region, it is crucial to track molten and water interface motion precisely.

Computing interface motion plays an important role in numerical multi-phase flows research. There are several methods developed for interface tracking or capturing in multi-phase Computational Fluid Dynamics (CFD) and different methods have their own characteristics. In general, the interface tracking method and interface capturing method are the two main classes of methods that are used to locate the interface. Both the classes are validated against a Taylor bubble benchmark problem by Marshall *et al.*[2].

Interface tracking methods includes the arbitrary Lagrangian-Eulerian (ALE) method and the marker and cell (MAC) method. A typical ALE method is front tracking method [3, 4], which is based on an adaptive mesh that deforms with the interface that is advected in a Lagrangian fashion by the velocity interpolated from the CFD velocity field. MAC method use a series of points inside a fixed-grid region to represent the interface. The points are moved in Lagrangian fashion in the velocity field solved from Navier-Stokes equations. Interface tracking method can precisely describe the motion of interface but can not strictly conserve mass and the algorithm can be very sophisticated and time-consuming when the interface is deeply twisted. And this method is still under development and improvement. Marić *et al.* implemented a compromising hybrid Level Set/Front tracking Method for unstructured grids within OpenFOAM framework [5].

The interface capturing methods as another approach to simulate free surface and interfacial flows mostly include the volume of fluid(VOF) method[6, 7] and the level set(LS) method[8, 9], which both adopt a scalar function to capture the interface in fixed Eulerian grids. Currently the interfacial flow solvers take variants VOF methods as the interface advection step in most CFD codes, which include current versions of ANSYS Fluent®, STAR-CCM++®,OpenFOAM® and so on. The VOF method defines void fraction α as the fraction of volume occupied by the liquid in each cell. The void fraction α bounded between 0 and 1 at the cells fully occupied by one of phases, changes discontinuously across the interface. The most crucial step of the method is to solve the advection equation of α. But due to the discontinuous nature of α, large numerical diffusion in the convection scheme causes non-physical smearing of the interface. In order to describe the interface precisely, most VOF methods are separated into two categories, geometric VOF [6, 10–16] and algebraic VOF [7, 17–21]. Geometric VOF methods define or reconstruct the interface location within a cell using the function value α,

while algebraic VOF methods use high solution schemes and compressive to reduce the numerical diffusion without defining exactly the location of interface.

Generally, the interface described by the geometrical methods is more accurate and less smeared than the algebraic methods. Because in most geometrical methods, the interface segment is reconstructed within a cell volume by piecewise lines. During the past decades, scientists proposed variants geometrical methods, which include donor-acceptor[6], SLIC(simple line interface calculation)[12] and PLIC(piecewise linear interface calculation)[14]. The PLIC method is a second-order algorithm and the reconstructed interface is much shaper than other geometrical methods.The article[7] summaries the three geometrical methods and compares them with FCT(flux-corrected transport)-VOF method proposed in the article. The conclusion is that Youngs' method[14] may be more accurate but more complicated to apply in three dimensions and unstructured meshes than FCT-VOF methods. To be applied for three dimensions, the piecewise-linear interface calculation method is proposed in [11]. And the interface reconstruction method is realized in [10] for three dimensional arbitrary convex cells. One of the geometrical algorithms called isoAdvector is proposed in [15] and implemented into OpenFOAM® as the incompressible two-phase flow solver, isoAdvector. This method uses the concept of isovalue contours to locate the interface in cells. For full details, the source code is provided with this website[22].

Among algebraic methods, using standard upwind scheme to transport the VOF field is the easiest one to apply. Most algebraic methods use the compressive algorithms to discretize the convective term in the VOF advection equation for preserving the interface sharpness. The interface typically spreads over a few cells and diffuses lowly but unboundedly under certain conditions[18]. The interface smearing can be minimized and bounded by adding a compressive term or counter gradient term to the VOF advection equation[19]. Some solvers like interFoam and multiphaseInterFoam in OpenFOAM® use the algebraic methods, also called Multidimensional Universal Limiter with Explicit Solution (MULES) algorithm, which is highly adaptive to three dimensions and unstructured grids. The performance of the two-phase flow solver - interFoam is evaluated in [21] and the basic principle of MULES algorithm can also be found.

Level set method was first proposed by Osher and Sethian [23], and then it was coupled to the equations for two-phase incompressible flow by Sussman *et al.*[8]. LS method was proved to be a powerful algorithm to handle complex topological changes such as merging, twisting and pinching by following research [9, 24, 25]. And exhaustive explanation and general applications of LS method can be found in Osher and Fedkiw's book[26] and Sethian's book[27]. In stead of using bounded volume fraction, the idea of LS method is to represent the interface with the zero value iso-face of a level set function ϕ. The value of ϕ is positive in one phase and negative in the other [28]. One advantage of LS method is that the interface is advected implicitly by solving the advection equation of ϕ, which solved algebraically with high order discretization schemes like WENO (weighted essentially non-oscillatory) scheme [29–31]. Due to the Lipschitz-continuous nature of the level set function, which is usually takes the form of the signed distance to the interface, the derivatives of ϕ are easy to calculate as well as the normal and curvature of the interface [32]. The signed distance function prevents gradients of ϕ from being steep and flat as that can cause numerical instabilities and loss inaccuracy. As ϕ is advected by solving the transport equation, the interface's shape is changed and the level set function loses the characteristic of the signed distance function that can be restored by reinitialization equation[8, 33]. However, mass loss/gain always occurs along with the simulation of incompressible flows, because LS function can not provide volume information as VOF does. Furthermore, numerical errors arise from solving the LS advection equation and/or the reinitialization equation. Especially in areas of high curvature or other unsolved regions, the discretization of advection equation can cause inevitable and significant numerical dissipation[34]. Many attempts have been made to improve the mass conservation of the level set method and reduce numerical dissipation from advection equation and reinitialization process. This article [35] summarized several glories of improvement strategies.

A lot of studies were devoted to improve level set method. Some applied high solution schemes like fifth-order WENO scheme[36, 37], discontinuous Galerkin method [38–40] and semi-Lagrangian approach[41–43]. Some articles studied velocity extensions method to maintain the signed distance function[44–47]. Moreover, using hyperbolic tangent function to supersede the signed distance function as the level set function was firstly proposed by Olsson *et al.*[48, 49]. A series of work based on the method has been done[50–59]. But small pieces of fluid are found breaking off from the interface and moving with erroneous velocity field[35]. Besides, Kohno *et al.* [60] proposed a novel numerical method for solving the advection equation for a level set function by using hierarchical-gradient truncation and remapping (H-GTaR) of the original PDE. In addition, spatially adaptive level set methods were developed to improve the accuracy of the interface location. These methods include the adaptive level set approach[61], the octree based methods [34, 62], the structured adaptive mesh refinement[36], the refined level set grid (RLSG) method [63], the spectrally refined interface approach [64] and the adaptive level set method[65]. These articles[66–69] combined LS method with Ghost FLuid Method [70] and the boundary conditional capturing technology[71]. On the other hand, Some studies are aimed for improving the accuracy and efficiency of the reinitialization process[25, 72–74]. There are several ways to restore the signed distance function, including PDE (partial differential equation) method [9, 75, 76] and FMM (fast marching method) [59, 64, 77–80]. Besides, more sophisticated methods for reinitialization have been proposed, including geometric mass-preserving redistancing scheme [81] and the volume-reinitialization scheme[82].Hysing

and Turek discuss and compare these reinitialization methods in [83].

To take advantages of both LS and VOF methods, Sussman *et al.*[84, 85] proposed a coupled level set and volume of fluid (CLSVOF) method. These studies applied CLSVOF method and its variants[86–91]. Some improved CLSVOF methods, like VOSET method[92],the conservation correction equation method[93], and the level set volume constraint method[94] were developed. Cao *et al.*[95] applied VOSET method based on multi-dimensional advection for unstructured triangular meshes. Pijl *et al.* [96, 97] proposed a mass-conserving method, which does not need to construct the complicated interface. Besides, these articles[98–101] applied the hybrid Lagrangian-Eulerian method to rebuild the level set function.

In this paper, we introduce the CLSAdvection algorithm which is a mass conservative level set method coupled with volume of fluid method. The purpose of developing this algorithm is to combine the continuous nature of level set function ϕ and the mass conservative property of void fraction α. CLSAdvector algorithm significantly reduces the smearing of the interface and avoids the mass loss caused by numerical diffusion. This algorithm is developed into a multi-phase solver implemented in the OpenFOAM® CFD software.

In the reminder of this section, we review the interface problem and all kinds of previous studies on the interface tracking/captring methods. In section 2, we introduce the concept of CLSAdvector algorithm and give an overview of the steps involved in the numerical procedure. In section 3, the corresponding numerical implementations are presented in detail. This is followed by section 4, in which four benchmark case are tested for validation. Finally, some conclusions are drawn in section 5.

2 The CLSAdvection concept

The idea of CLSAdvector was inspired by Roenby *et al.* who proposed isoAdvector concept[22], which is a geometric VOF method. In the isoAdvector method, void fraction α is interpolated into the vertexes of the grids. Generally, it is customary to visualize the interface by showing the 0.5-isosurface based on the void fraction α with the post-processing procedure, like ParaView®. However, it is not precise to use 0.5-isosurface as interface because it can hardly cut the cell into two subcells of the volumetric proportions dictated by the volume fraction α. So the most important step of isoAdvector is to find the exact value of α that can cut the surface cells into two subcells with right void proportions. This method is quite novel and it is applied to unstructured meshes and implemented as an incompressible two-phase flow VOF solver in OpenFOAM®. However, isoAdvector still has the same faults as other geometric VOF methods, in which the curvature and normal direction of interface are influenced by the discontinuous property of α. Different from the isoAdvector method, CLSAdvector uses the projection distance D of $\vec{x}_j$ (the vector from cell center to i-th vertex) on interface normal direction $\vec{n}$ to reconstruct the interface that meets void fraction α. For the sake of clarity, only ideas about CLSAdvector are focused in this section, and the numerical detailed description of the implementations can be found in section 3. The reader is referred to the source code provided at this website[102].

2.1 Governing equations for incompressible two phase flow

The incompressible Navier-Stokes equations for two immiscible phases(A and B) read

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot (2\mu D) + \mathbf{f_\Gamma} + \mathbf{g} \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2}$$

where $\mathbf{u}$ is the velocity, ρ is the density, p is the pressure, $\mathbf{g}$ is the gravitational acceleration and μ is the dynamic viscosity. The material properties are constant for each phase and discontinuous at the interface. To avoid numerical instability [30], it is needed to make fluids properties in a continuous form as follows

$$\rho = \rho_A + (\rho_B - \rho_A)H(\phi), \tag{3}$$

and

$$\mu = \mu_A + (\mu_A - \mu_B)H(\phi). \tag{4}$$

In equation 1, D is defined as the rate of deformation tensor,

$$D = \frac{\nabla \mathbf{u} + \nabla \mathbf{u}^T}{2}. \tag{5}$$

$\mathbf{f_\Gamma}$ is the surface tension force, which is transformed to be a volume force scattering within the interface region by CSF (continuum surface force) model[103] as follows

$$\mathbf{f_\Gamma} = \frac{\sigma \kappa \nabla H_\varepsilon(\phi)}{\rho_\varepsilon(\phi)}. \tag{6}$$

At the interface Γ, separating the two fluids, there is the normal continuity condition for velocity

$$u_{A,\perp} - u_{B,\perp} = \mathbf{u}_A \cdot \mathbf{n} - \mathbf{u}_B \cdot \mathbf{n} \equiv 0, \tag{7}$$

the tangential continuity condition for velocity

$$u_{A,\parallel} - u_{B,\parallel} \equiv 0, \tag{8}$$

and the jump condition for surface stress

$$[\mathbf{n} \cdot (-p\mathbf{I} + 2\mu D) \cdot \mathbf{n}] = \sigma\kappa, \tag{9}$$

where $\mathbf{I}$ is the identity matrix, σ is the surface tension and κ is the interface curvature. According to the article[96], if viscosity is continuous across the interface, the coupled jump condition of pressure and velocity gradient(9) can be decoupled. Then we can get

$$[\nabla \mathbf{u}]_\Gamma = 0, \tag{10}$$

and

$$[p]_\Gamma = \sigma\kappa. \tag{11}$$

The curvature κ is calculated from

$$\kappa = -\nabla \cdot \mathbf{n} = -\nabla \cdot \frac{\nabla\phi}{|\nabla\phi|}, \tag{12}$$

where $\mathbf{n}$ is the unit normal vector. The jump conditions for pressure and materials are smeared out over the interface region of a thickness $\eta = 2\varepsilon$. A smoothed Heaviside function is applied to make the jump conditions smeared and continuous as follows

$$H_\varepsilon(\phi) = \begin{cases} 0, & \phi < -\varepsilon \\ \frac{1}{2}[1 + \frac{\phi}{\varepsilon} + \frac{1}{\pi}\sin(\pi\frac{\phi}{\varepsilon})], & |\phi| \le \varepsilon. \\ 1, & \phi > \varepsilon \end{cases} \tag{13}$$

2.2 Interface reconstruction

The kernel of CLSAdvector algorithm is to reconstruct interfaces in surface cells with normal direction $\mathbf{n}$ calculated from ϕ and void fraction α. First of all, it is necessary to define the volume fraction α of fluid phase A in cell i at time t,

$$\alpha_i(t) \equiv \frac{1}{V_i}\int_{\mathscr{C}_i} \Pi(\mathbf{x}, t)dV \tag{14}$$

where Π is a characteristic function,

$$\Pi(\mathbf{x}, t) \equiv \begin{cases} 1, & \mathbf{x} \in PhaseA \\ 0, & \mathbf{x} \in PhaseB. \end{cases} \tag{15}$$

Although the interface may be actually curved inside a surface cell, it is approximately regarded as a plane inside the cell. In order to locate the position of interface with the two important variables $\mathbf{n}$ and ϕ in a polyhedral cell, in CLSAdvection method, iso-value surfaces which can cut the cells into the correct ratio are calculated. In order to find this iso-value for a given surface cell, an efficient method is proposed and the details can be seen at section 3. We have to admit that the iso-value faces are not continuous because they are designed to meet the mass conservative condition as we can see in figure 1. Therefore, the reconstructed interfaces are only used to calculate the void fraction field of the next time step by simulating the motion of the interface, while the normal direction $\mathbf{n}$ is provided by solving the advection of level set function.

2.3 Interface motion

The following equation for the volume of fluid is solved to keep the free surface conservative,

$$\frac{\partial\alpha}{\partial t} + (\mathbf{u} \cdot \nabla)\alpha = 0. \tag{16}$$

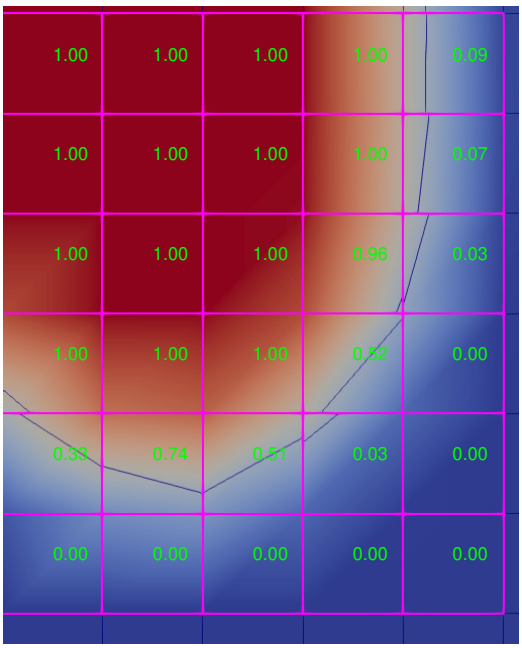


Figure 1: Noncontinuous reconstructed interfaces

As the system of all governing equations is solved in sequence within a time step, only the information about velocity field U, void fraction α and level set function ϕ at time t is available. The void fraction field of the next time step is calculated by the following function,

$$\alpha_i(t+\Delta t) = \alpha_i(t) - \frac{1}{V_i}\sum_{j\in\mathscr{B}_i}\sigma_{ij}\int_t^{t+\Delta t}\int_{\mathscr{F}_i}\Pi(\mathbf{x},\tau)\mathbf{u}(\mathbf{x},\tau)d\mathbf{S}d\tau, \tag{17}$$

where $\sigma_i j \in +1,-1$, such that $\sigma_{ij}d\mathbf{S}$ on each cell face points from cell i to cell j. The information between the interval $[t, t+\Delta t]$ is unknown. Hence it is simply estimated that the velocity field is regarded as constant during the time step. Besides, the velocity $\mathbf{u}(\mathbf{x},t)$ on the cell face is dotted with differential face normal vector, $d\mathbf{S}$, which is difficult to be integrated because of the unknown distribution of velocity on the cell face. This term can be approximated as average velocity, $u_{ij}(t)$, multiplied with differential face as follows:

$$\mathbf{u}(\mathbf{x},t)d\mathbf{S} \approx u_{ij}(t)dS = \frac{F_{ij}(t)}{S_{ij}}dS, \tag{18}$$

where volumetric face flux, F_{ij}, is defined to estimate the average velocity on the cell faces,

$$F_{ij}(t) \equiv \int_{\mathscr{F}_{ij}}\mathbf{u}(\mathbf{x},t)d\mathbf{S}. \tag{19}$$

Substitute this into (30) and we can have

$$\alpha_i(t+\Delta t) = \alpha_i(t) - \frac{1}{V_i}\sum_{j\in\mathscr{B}_i}\sigma_{ij}\frac{F_{ij}(t)}{S_{ij}}\int_t^{t+\Delta t}\int_{\mathscr{F}_i}\Pi(\mathbf{x},\tau)dSd\tau. \tag{20}$$

The remaining face integral means the area submerged in fluid A at time t, $\mathscr{A}_{ij}(t)$,

$$\mathscr{A}_{ij}(t) \equiv \int_{\mathscr{F}_i}\Pi(\mathbf{x},t)dS. \tag{21}$$

The volume of fluid A across each cell face in time interval $[t, t+\Delta t]$ can be approximated with following equation,

$$\Delta V_{ij}(t,\Delta t) \approx \frac{F_{ij}(t)}{S_{ij}}\int_t^{t+\Delta t}\mathscr{A}_{ij}(\tau)d\tau. \tag{22}$$

As above equations show, in order to calculate the void fraction $\alpha_i(t+\Delta t)$ at the next time step, it is necessary to calculate the area submerged by the reconstructed interface during Δt and summarize the volume of fluid A, ΔV_i across each face of the cell. The algorithm of estimating the interface motion was proposed by Johan Roenby *et al.* and can be found in [15].

2.4 Level set function initialization

At time t, the void fraction field $\alpha(t)$ is known. In volume of fluid method, the normal direction $\mathbf{n}$ is calculated by following equation,

$$\mathbf{n} = \frac{\nabla\alpha}{|\nabla\alpha|}. \tag{23}$$

However, α field has discontinuous characteristic, $\alpha \in [0, 1]$, which can lead to large numerical diffusion and non-physical smearing of the interface. Hence it is difficult to obtain high precision normal direction and curvature with VOF method especially for unstructured grid [95]. To improve the interface capturing method, level set function ϕ with continuous characteristic at the interface is used in CLSAdvection method. The normal direction of interface is defined as follows,

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}. \tag{24}$$

The level set function field $\phi(t)$ at time t can be obtained from the volume fraction field $\alpha(t)$ with following steps. First, the interface cells are found with the condition that $\alpha_i \in (0, 1)$ and signed with flag 0. Secondly, find its first layer around the interface cells and sign all the first layer cell with flag 1. Do the same with second layer cells and sign them with flag 2. Both the layers limit the thickness of the interface and ensure its precision. Thirdly, it is assumed that the continuous interface is the iso-surface at $\alpha = 0.5$ and initialize the ϕ with following equation,

$$\tilde{\phi}(t) = 1.5\varepsilon(2 * \alpha(t) - 1). \tag{25}$$

Apparently, the first initialized level set function $\tilde{\phi}$ is not the signed distance function. There are lots of methods to re-initialize the level set function, including partial differential equation method (PDE) and fast marching method (FMM). PDE method can be easily realized in unstructured grid for it is only need to solve the following re-initialization equation, which is also a hyperbolic equation,

$$\frac{\partial\phi}{\partial\tau} + sgn(\tilde{\phi})(|\nabla\phi| - 1) = 0, \tag{26}$$

in which $sgn(\phi)$ is a sign function as following,

$$sgn(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1. & x < 0 \end{cases} \tag{27}$$

In order to solve equation (26), Godunov's method for discretizing the hyperbolic term $sgn(\tilde{\phi})\,|\nabla\phi|$, is recommended in [26]. After solving the re-initialization equation, the signed distance field ϕ is obtained. Nonetheless, it still need correction to ensure the mass conservative characteristic. At last, our interface can be implicitly expressed with level set function, which is confined in a band containing the interface to save computing resource. All the detailed numerical algorithms can be found in section 3.

2.5 Normal direction and curvature

It is already known that level set method has the advantage of computing accurate interface normal direction but the disadvantage of keeping mass conservation. Hence, the signed distance level set function ϕ is used to calculate the normal direction at the next time step $t + \Delta t$ by solving the following hyperbolic convective equation,

$$\frac{\partial\phi}{\partial t} + (\mathbf{u} \cdot \nabla)\phi = 0. \tag{28}$$

Equation (28) is sometimes referred to as the *level set equation*, which was firstly proposed by Osher and Sethian [23]. After the equation is discretized with the finite volume method (FVM), it is essential to choose a suitable scheme to compute the fluxes of ϕ at all boundary faces of interface cells[31] .Because the formation of large gradients during the interface motion can cause spurious oscillations near discontinuities at the interface and lose of the interface sharpness[104]. Using an arbitrarily Weighted Essentially Non-Oscillatory (WENO) scheme proposed by Pringuey and Cant [105] can handle complex interface. WENO schemes have the ability to preserve the required sharpness of the interface in front-propagating problems and to deal with large gradients with discontinuities. [106] takes the advantage of WENO scheme to solve equation(28) and realize the coupled level set and volume of fluid in OpenFOAM® with a hyperbolic tangent function ψ, which is similar to Heaviside function, H in equation(13). Martin and Shevchuk[31] apply semi-implicit WENO schemes using OpenFOAM® and the source code is open source used in this article's work. Besides, a third-order Total Variation Diminishing (TVD) Runge-Kutta(RK) scheme for temporal discretization is used for the advection steps[107].

2.6 Algorithm overview

The numerical algorithm combines the method of volume of liquid and level set. The solver start off with the initialization of velocity, pressure and α fields. Before the time loop, the LS variable, ϕ, generate from the α field and create other variables like δ and H. Then PIMPLE loop starts from the interface reconstruction with α and $\vec{n}$. The figure (2) shows the specific flow chart.

3 Numerical implementations

3.1 Cutting the cell with α and ϕ

In this case, the normal direction $\mathbf{n}$ of certain cell is given by level set function ϕ and the void fraction α is given by volume of fluid function. This section means to explain the algorithm of finding such a plane with normal direction $\mathbf{n}$ that can cut the cell into the right void fraction α. The locations of cell center point $\mathbf{x_i}$ and N cell vertexes $\mathbf{x_{i_1}}, ..., \mathbf{x_{i_N}}$ are needed. Then we can get N vectors $\mathbf{d_1}, ..., \mathbf{d_N}$ from cell center to N cell vertexes. The projections of center point to vertex on the normal direction can be calculated as

$$D_i = \mathbf{d_i} \cdot \mathbf{n}, \quad for \quad i = 1, ..., N. \tag{29}$$

Let us suppose the objective plane contain one of the vertexes and we can have a series of plane to cut the cell into different fractions (figure 3.1). Due to the cell's polyhedral characteristic, a piecewise function about the center cell distance and void fraction is drew in figure 3.2. The objective plane with the given void fraction α_i must have a certain distance D^* to the center point $\mathbf{x_i}$ such that $\tilde{\alpha}(D^*) = \alpha_i$ .The first step is to find the point on the certain part of the piecewise function, which can be realized by comparing the void fraction value of vertexes and α^*. Suppose the point p is between point k and l, such that $D^* \in [D_k, D_l]$. We use a cubic polynomial to fit this interval. The second step is to find the two trisection point in this interval, say, m and n, and calculate the $\tilde{\alpha}(D_m)$ and $\tilde{\alpha}(D_n)$ in geometric way. Then we have four points for the four polynomial coefficients by solving a group of linear equations. Use LU decomposition to solve the linear 4×4 Vandermonde matrix system. Then use Newton's root finding method to find D^* with the condition, $|\tilde{\alpha}(D^*) - \alpha_i| < \epsilon$. ϵ is a user-defined tolerance, typically set to 10^{-8}.

3.2 Narrow band containing interface

The cells that contain the interface are limited to where $H \in (\xi, 1 - \xi)$. ξ is defined by users to confine the algorithm resolution, usually set as 0.0001. The reason why uses Heaviside function rather than void fraction is that the interface defined by Heaviside function is more explicit and void fraction function more smeared. However, only interface cells are not enough to reconstruct the details of the interface. Level set method needs to build the signed distance function in the whole field, which can precisely capture the interface. Nonetheless, it takes too much computational resource and time to calculate on the whole field. So it is necessary to build a narrow band that has a thickness of two layers of grid like figure(4). The first step is to sign all the interface cells with the flag "seed" and all the non-interface cell with the flag "away". The second step is to find the cells that share the vertexes of the "seed" cell. If the cells are signed with flags including "away" or "second layer", sign them with the flag "first layer". The third step is to traverse the cells that share the vertexes of the "first layer" cells. If the cells are signed with flags "away", sign them with the flag "second layer". Thus the narrow band that contains the interface and two-layers grids is built. The Heaviside function field and normal direction vector field can be limited in the narrow band to evade unnecessary computation.

3.3 Reinitialization

Reinitialization is actually the process of substitute $\tilde{\phi}(x,t)$ with another function $\phi(x,t)$ that has the same zero contour as $\tilde{\phi}(x,t)$ but the better property, $|\nabla\phi| = 1$[75]. There are two ways to realize the process, including a direct, fast marching method(FMM)[27] and PDE-based method [8] by converting it into a time-dependent Hamilton-Jacobi equation 26. As Sussman and Osher *et al.* suggested in article [8], the sign function $sgn(x)$ can be approximated as the following equation,

$$sgn(\tilde{\phi}) = \frac{\tilde{\phi}}{\sqrt{\tilde{\phi}^2 + \epsilon^2}}, \tag{30}$$

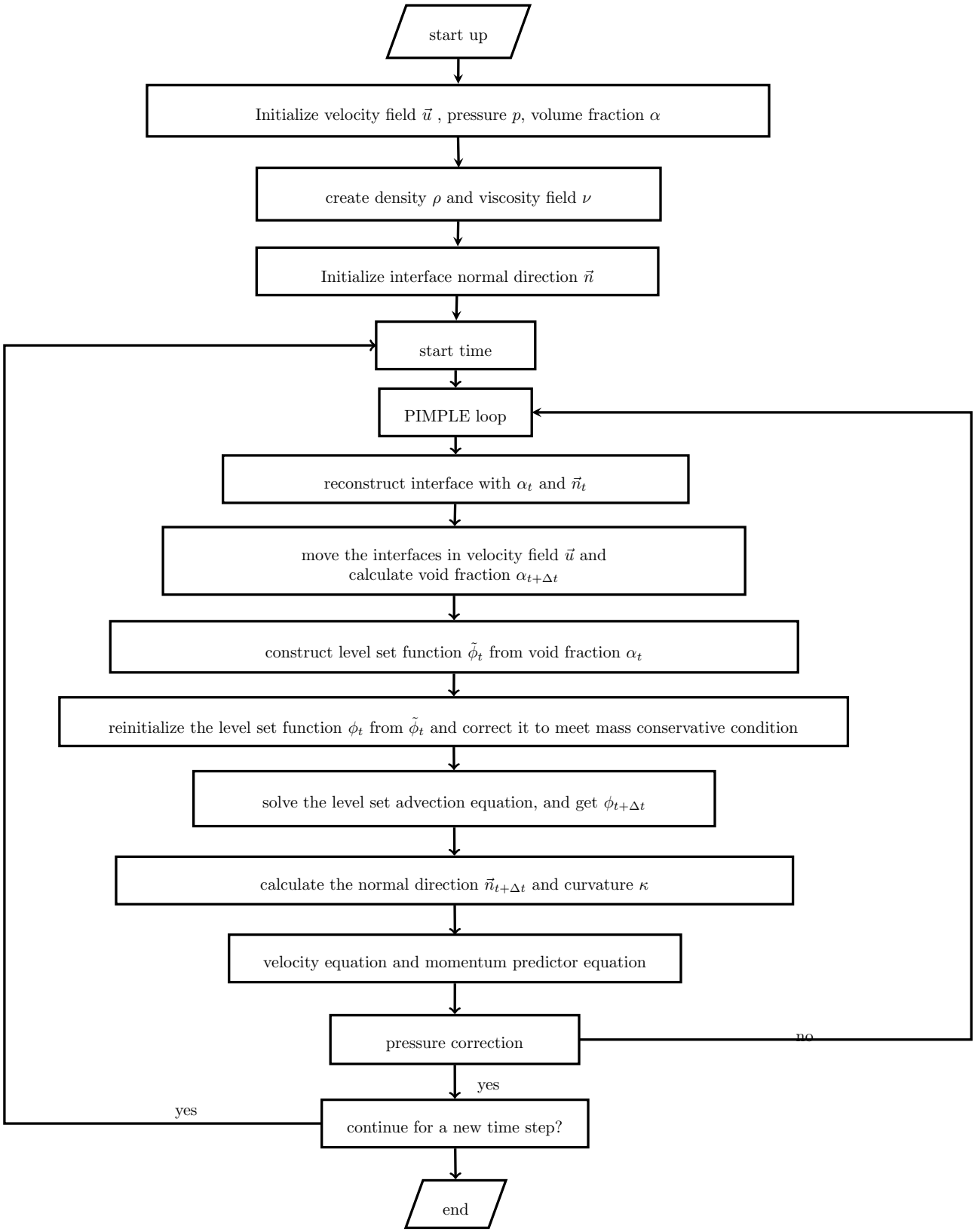
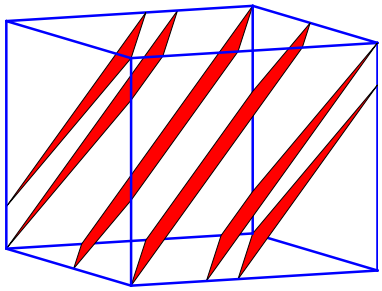
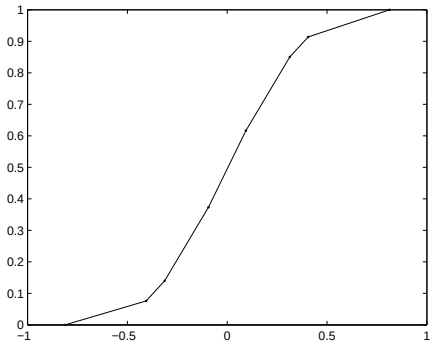


Figure 2: Algorithm for the CLSAdvection solver.



3.1



3.2

Figure 3: 3.1 shows that planes pass different vertexes with the same normal direction. 3.2 shows the cut volume and distance to the plane.

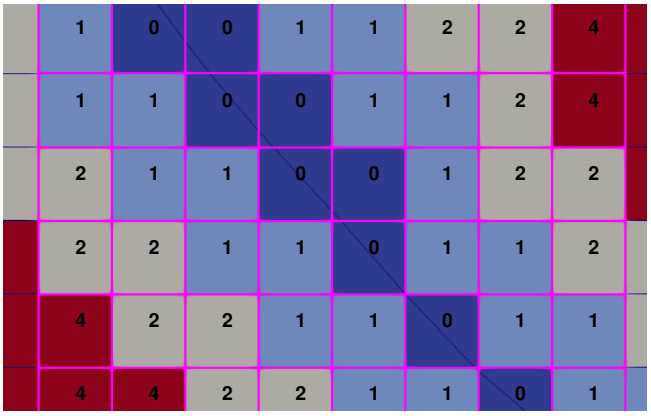


Figure 4: The numbers in cells mean the flag. 0: "seed", 1:"first layer", 2:"second layer" and 4:"away"

where ϵ means the thickness of the interface in this article. To solve the Hamilton-Jacobi equation (26),with $sgn(\phi)$ approximated by (30), there is a canonical monotone upwind scheme, Godunov's scheme, described in [26, 108] and the equation is discretized as the following,

$$\phi_{i,j}^{n+1} = \phi_{i,j}^{n} - \Delta\tau sgn_{\epsilon}(\tilde{\phi})G(\phi_{i,j}^{n}), \tag{31}$$

where $sgn_{\epsilon}(\tilde{\phi})$ is the approximation to equation(30) and $G(\phi_i^n)$ is defined as

$$G(\phi_i^n) = \begin{cases} \sqrt{max[(l^+)^2,(r^-)^2] + max[(s^+)^2,(n^-)^2] + max[(b^+)^2,(t^-)^2]} - 1 & \tilde{\phi}_i > 0 \\ \sqrt{max[(l^-)^2,(r^+)^2] + max[(s^-)^2,(n^+)^2] + max[(b^-)^2,(t^+)^2]} - 1 & \tilde{\phi}_i < 0 \\ 0 & others \end{cases} \tag{32}$$

The parameters in equation(24) show as follows,

$$\begin{aligned}
l &= \sum_{j\in\mathsf{F}} D_{i,j} \max[\frac{\vec{x}_j - \vec{x}_i}{|\vec{x}_j - \vec{x}_i|} \cdot (-1,0,0), 0] \\
r &= -\sum_{j\in\mathsf{F}} D_{i,j} \max[\frac{\vec{x}_j - \vec{x}_i}{|\vec{x}_j - \vec{x}_i|} \cdot (1,0,0), 0] \\
s &= \sum_{j\in\mathsf{F}} D_{i,j} \max[\frac{\vec{x}_j - \vec{x}_i}{|\vec{x}_j - \vec{x}_i|} \cdot (0,-1,0), 0] \\
n &= -\sum_{j\in\mathsf{F}} D_{i,j} \max[\frac{\vec{x}_j - \vec{x}_i}{|\vec{x}_j - \vec{x}_i|} \cdot (0,1,0), 0] \\
t &= \sum_{j\in\mathsf{F}} D_{i,j} \max[\frac{\vec{x}_j - \vec{x}_i}{|\vec{x}_j - \vec{x}_i|} \cdot (0,0,-1), 0] \\
b &= -\sum_{j\in\mathsf{F}} D_{i,j} \max[\frac{\vec{x}_j - \vec{x}_i}{|\vec{x}_j - \vec{x}_i|} \cdot (0,0,1), 0] \\
D_{i,j} &= \frac{\phi_{i,j} - \phi_i}{|\vec{x_j} - \vec{x}_i|}, j \in \mathsf{F} \\
z^+ &= max[z,0], z^- = min[z,0],
\end{aligned} \tag{33}$$

where F means the cells that share faces with cell i. The above algorithm uses the Godunov's scheme in unstructured grid, which is successfully applied in OpenFOAM®. During the calculation, the number of the iteration steps are defined by users and controlled by two input parameters, "CFL" and "ENUM" in the "tranportProperties" dictionary. The max iteration number NITER is defined as $NITER = CFL \times ENUM$. Actually, the optimal step number is 10 and the level set function can reach a satisfied effect according to [29].

3.4 Correction for mass conservation

The continuous interface is firstly assumed at the iso-surface at $\alpha = 0.5$. Then level set function is built to simulate the interface is at $\phi = 0$. After the reinitialization process, the level set function needs to be corrected for the process slightly change the position of the interface to meet the mass conservative condition. The correction procedure takes advantage of cubic polynomial function to fit the relation of ϕ and total mass M. Dur to the narrow band, level set function ϕ is limited at $[\min(\phi), \max(\phi)]$. The first step of correction is to calculate the value of three points,$0.1\max(\phi), 0.05\max(\phi), 0.05\min(\phi)$. Secondly, calculate the volume wrapped by the iso-value contours including the three points and zero points. Then there are four interpolation values for getting the cubic polynomial coefficients by solving a linear equation group. The detailed algorithms including LU decomposition and Newton's root finding method are the same as section 3.1 of how to getting the reconstructed interface position. After correction, the continuous, mass-conservative interface is obtained and ready for convection in the velocity field.

3.5 Numerical Discretization of level set equation

In some articles like [92, 109], the level set equation is not solved. And the level set function produced from the α field. However, it is testified in this article that not-solving the level set equation explicitly increase the distortion of the interface, because only the advection of void volume is not enough to calculate the normal direction at the next time step, let alone predict the position of the interface precisely. In this article, a third-order TVD-RK temporal scheme and semi-implicit third-order upwind WENO scheme or Vanleer scheme are used in solving the level set equation (28).

$$
\begin{aligned}
\phi^{(0)} &= \phi_t \\
\phi^{(1)} &= \phi^{(0)} - \Delta t(\vec{u}\cdot\nabla)\phi^{(0)} \\
\phi^{(2)} &= \frac{3}{4}\phi^{(0)} + \frac{1}{4}\phi^{(1)} - \frac{1}{4}\Delta t(\vec{u}\cdot\nabla)\phi^{(1)} \\
\phi_{t+\Delta t} &= \frac{1}{3}\phi^{(0)} + \frac{2}{3}\phi^{(2)} - \frac{2}{3}\Delta t(\vec{u}\cdot\nabla)\phi^{(2)}
\end{aligned}
\tag{34}
$$

The discretization of $(\vec{u}\cdot\nabla)\phi^{(0)}$ is easily to realize because the WENO scheme and Van Leer scheme are already coded in OpenFOAM®, which are not the main work of this article and only need to add the source files and correctly write the dictionary, "fvScheme".

4 Results

In the following, some test cases with CLSAdvection method are presented. It is known that the level set method can improve the sharpness of the interface but also cause mass loss. Whereas, the volume of fluid method keeps the interface mass conservative but smeared. Therefore, the validation of the CLSAdvection method is tested in this article's work and there are several simple cases comparing this coupled method with the level set method and volume of fluid method. The error measures will be used to quantify the solution quality with following aspects.

— Volume conservation

In order to track the precision of the advection process, volume conservation is measured to testify whether the simulated results conform to Navier-Stokes equations' requirement. The fractional volume conservative error is defined as following equation by this article[110],

$$
\varepsilon_V(t) = \frac{\left|\sum_{i=1}^{N}\alpha_i(t)V_i - \sum_{i=1}^{N}\alpha_i(0)V_i\right|}{\sum_{i=1}^{N}\alpha_i(0)V_i}. \tag{35}
$$

— Mass conservation

The introduction of level set method that can cause the loss of mass for using equation (3) to define the density may underlie a slight mass loss. Nevertheless, the coupled method is aimed to improve the algorithm's ability of mass conservation, which is measured by the following equation,

$$
\varepsilon_M(t) = \frac{\left|\sum_{i=1}^{N}\rho_i(t)V_i - \sum_{i=1}^{N}\rho_i(0)V_i\right|}{\sum_{i=1}^{N}\rho_i(0)V_i}. \tag{36}
$$

— Sharpness

The interface is shaper, the width of the region where α changes from 0 to 1 is thinner. Therefore, the following equation measures the sharpness.

$$
\varepsilon_S(t) = \frac{\sum_j \alpha_j(t)V_j}{\sum_i \alpha_i(t)V_i}, \tag{37}
$$

where j means the cells where $0.01 \leq \alpha_j(t) \leq 0.99$.

— Boundedness

Volume fractions need to be physically meaningful, which means the condition $0 \leq \alpha \leq 1$ should be met. The measures of boundedness is $\min(\alpha)$ and $\max(\alpha)$. The measurements are taken all over the domain at the end of the calculation.

— Efficiency

Efficiency is measured by the simulation times T_{calc}. All the simulation were executed on a single core of an Intel® 2.80 GHz CPU(i7-7700HQ) on a Dell® Precision 3520 Mobile Workstation. We compare CLSAdvection's performance with an algebraic VOF scheme[21] and a geographic VOF scheme[15] to benchmark the algorithm. Both the VOF schemes are implemented in the OpenFOAM® and well developed for arbitrarily unstructured meshes.

4.1 Zalesak's test problem

Solid body rotation of a notched disc is commonly used to test the advection capabilities of an interface capture solver. Zalesak firstly introduced this test [111] and its variants have been used extensively. In this particular test, the radius of the disc is $R = 0.15$, the notch width is $W = 0.06$ and the notch height is $H = 0.25$. The center of the disc lies in $(x_0, y_0) = (0.5, 0.75)$ in a domain of 1×1 that is discretized using rectangular meshes of size 100×100,200×200,400×400 figure (5). The rotation velocity is given by the following equation,

$$\begin{aligned} u &= -2\pi(y - 0.5) \\ v &= 2\pi(x - 0.5). \end{aligned} \tag{38}$$

At all the domain boundary, zero gradient condition is set for α and ϕ.

4.1.1 structured meshes

In figure (6,7,8), the solutions of five combinations of time and mesh resolution in columns are obtained with MULES, isoAdvector, CLSAdvection. The effects of refining mesh resolution are investigated with fixed Courant Number, $Co = 0.5$. With the finest mesh, the effects of reducing Co from 0.5 to 0.1 are tested. Errors and Efficiency measures are displayed in table 1. Then we can get the following observations. All the algorithms are able to realize the volume conservation. But CLSAdvection method can cause slightly mass loss, which is confined at 0.01. Especially, CLSAdvection can get the same sharpness as isoAdvector which is a sharp interface advection method and far better sharpness than the MULES method. Both isoAdvector and CLSAdvection method can make sure the void fraction α bounded at $[0, 1]$. And MULES algorithm can cause a slight over flow. From table (1), we can see the coupled VOF and level set method take the most time to finish the cases, which is caused by solving reinitialization equation (26) and level set equation (28). Besides, from table (1), the influence of Courant number is that the smaller Co means smaller time steps and the longer simulation time. Nevertheless, shrinking Courant number doesn't mean to improve the precision. Actually, the errors can be accumulated as the simulations process. From the figure (8) and figure (7), the shape of the notched disc is kept well at $Co = 0.5$ and deforms more explicitly in the cases of $Co = 0.1$ and $Co = 0.2$. On the other hand, the results show that MULES method can cause numerical oscillations. As the meshes are densified, the oscillations become apparent, which can be explained by introducing high resolution convection scheme including vanLeer, SuperBee, QUICK and so on [21].

4.1.2 unstructured meshes

We apply such algorithm on unstructured meshes and the initial shapes in figure (9) and figure (13). Figure (10), (11) and (12)show calculation results of MULES, isoAdvector, CLSAdvection after a circle in polygon meshes. And figure (14), (15) and (16)show calculation results of MULES, isoAdvector, CLSAdvection after a circle in triangle meshes. The Co numbers are set as 0.1 and 0.5, while the meshes have three different resolutions with the cell number 51967 and 203965. The errors and calculation times are showed in table 2 and 3. From the figures and tables, we can make such conclusions as follows. First, the basic function of coupled level set and VOF method can be realized in unstructured meshes. The cell cut algorithm can be applied in polygon and triangle meshes. Second, the CLSAdvection can get the same sharpness effect as the sharp interface method, isoAdvector. MULES method can cause the interface smeared. After all, CLSAdvection inherits the idea of geometric volume of fluid method and explicitly reconstruct the interface. Third, polygon meshes are proved to have the better interface capturing capability than triangle meshes. Apparently, structured meshes undoubtedly have the best calculation characteristic. However, the nature of widely applied CFD softwares like OpenFOAM® and Fluent® requires the algorithm should be applied into unstructured meshes.

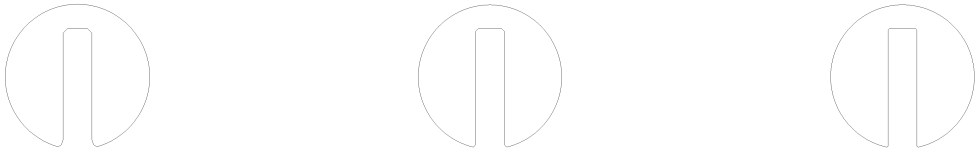
4.2 Dam Break

Here we test the CLSAdvection on the 2D dam break problem. This part we compare the CFD simulation results with the experiments made by Hu and Kashiwagi [**?**] for validation. The water tank with a removable partition is showed in figure **??**. The figure **??** shows the simulation results using the MULES, isoAdvector and CLSADvection methods and the experiment pictures.

4.3 Droplet

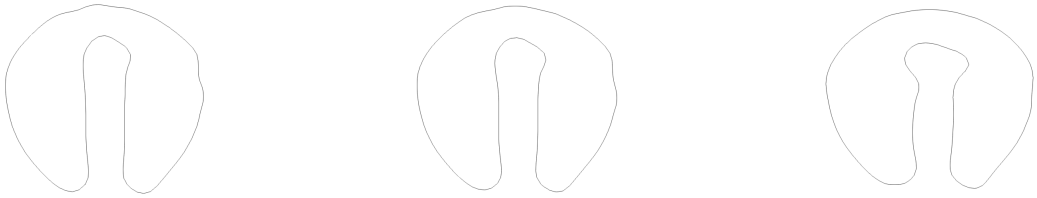
Here we test the CLSAdvection on the 2D droplet distortion problem. This part we compare the CFD simulation results with the CFD results made by Hu and Kashiwagi [**?**] for validation. The figure **??** shows the simulation results using the MULES, isoAdvector and CLSADvection methods and the experiment pictures.

5 Conclusions



5.1 5.2 5.3

Figure 5: Initial shape with different mesh sizes:5.1 100×100, 5.2 200×200, 5.3 400×400



6.1 6.2 6.3

6.4 6.5

Figure 6: The shape after one circle with MULES method:6.1 100×100 and $Co = 0.1$,6.2 100×100 and $Co = 0.2$,6.3 100×100 and $Co = 0.5$,6.4 200×200 and $Co = 0.5$,6.5 400×400 and $Co = 0.5$,

References

[1] T. Ginsberg, Liquid jet breakup characterization with application to melt-water mixing, Technical Report, Brookhaven National Lab., 1986.

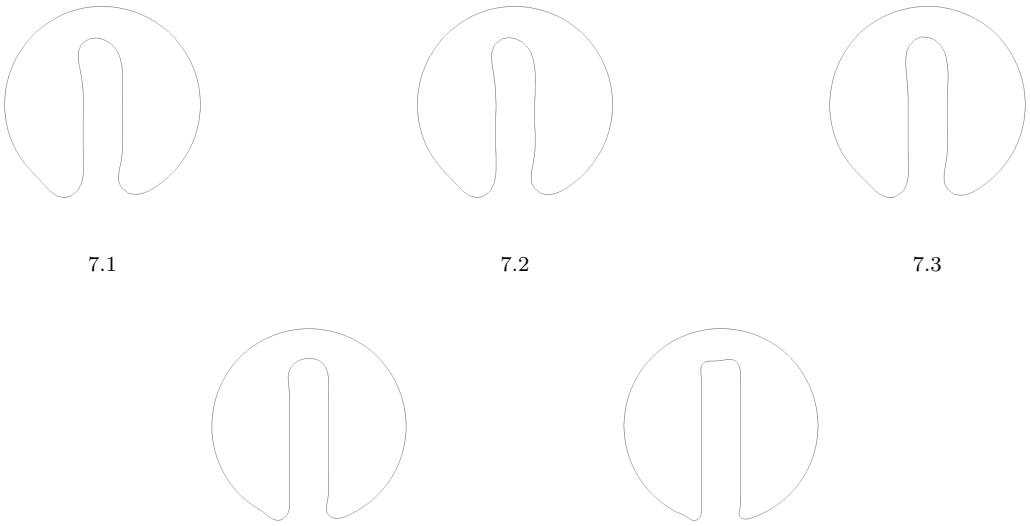
[2] H. Marschall, S. Boden, C. Lehrenfeld, U. Hampel, A. Reusken, M. Wörner, D. Bothe, et al., Validation of interface capturing and tracking techniques with different surface tension treatments against a taylor bubble benchmark problem, Computers & Fluids 102 (2014) 336–352.

[3] S. O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, Journal of Computational Physics 100 (1992) 25–37.

[4] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, Journal of Computational Physics 169 (2001) 708 – 759.

[5] T. Mari, H. Marschall, D. Bothe, lentfoam a hybrid level set/front tracking method on unstructured meshes, Computers & Fluids 113 (2015) 20 – 31. Small scale simulation of multiphase flows.

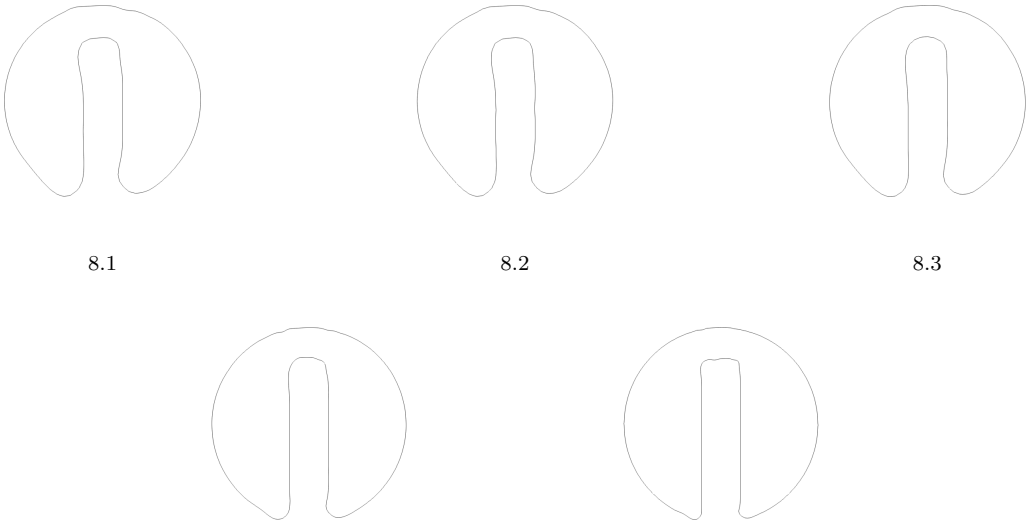
[6] C. W. Hirt, B. D. Nichols, Volume of fluid (vof) method for the dynamics of free boundaries, Journal of computational physics 39 (1981) 201–225.



7.1 7.2 7.3

7.4 7.5

Figure 7: The shape after one circle with isoAdvector method:7.1 100×100 and $Co = 0.1$,7.2 100×100 and $Co = 0.2$,7.3 100×100 and $Co = 0.5$,7.4 200×200 and $Co = 0.5$,7.5 400×400 and $Co = 0.5$,



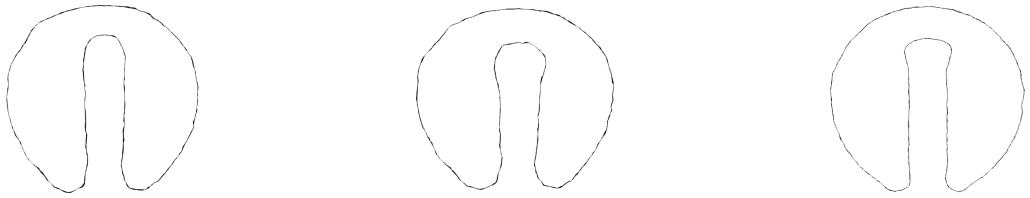
8.1 8.2 8.3

8.4 8.5

Figure 8: The shape after one circle with CLSAdvection method:8.1 100×100 and $Co = 0.1$,8.2 100×100 and $Co = 0.2$,8.3 100×100 and $Co = 0.5$,8.4 200×200 and $Co = 0.5$,8.5 400×400 and $Co = 0.5$,

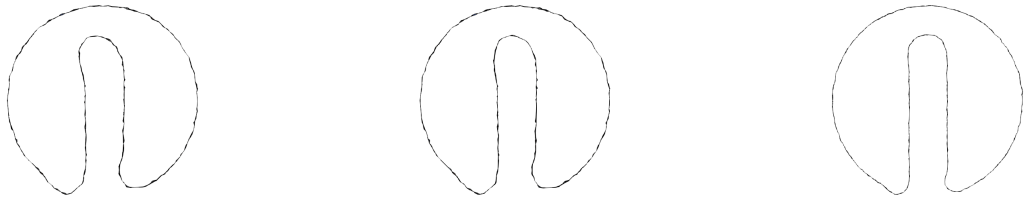
9.1 9.2

Figure 9: Initial shape in polygon meshes with different cell numbers:9.1 51967, 9.2 203965



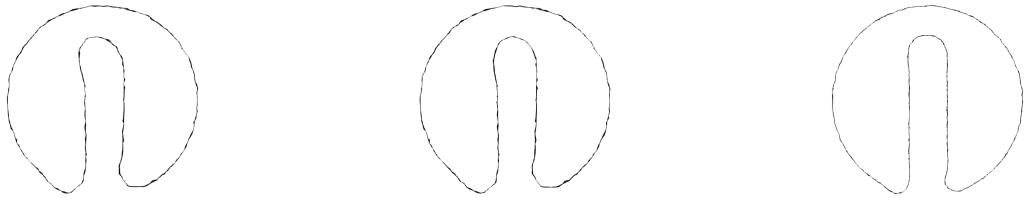
10.1 10.2 10.3

Figure 10: The shape after one circle with MULSE method:10.1 51967 and $Co = 0.1$, 10.2 51967 and $Co = 0.5$,10.3 203965 and $Co = 0.5$



11.1 11.2 11.3

Figure 11: The shape after one circle with isoAdvector method:11.1 51967 and $Co = 0.1$, 11.2 51967 and $Co = 0.5$,11.3 203965 and $Co = 0.5$

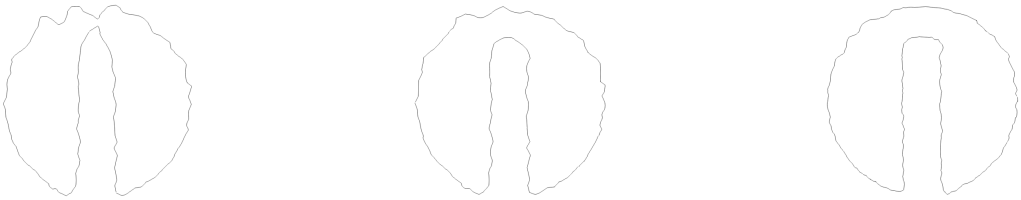


12.1 12.2 12.3

Figure 12: The shape after one circle with CLSAdvection:12.1 51967 and $Co = 0.1$, 12.2 51967 and $Co = 0.5$,12.3 203965 and $Co = 0.5$

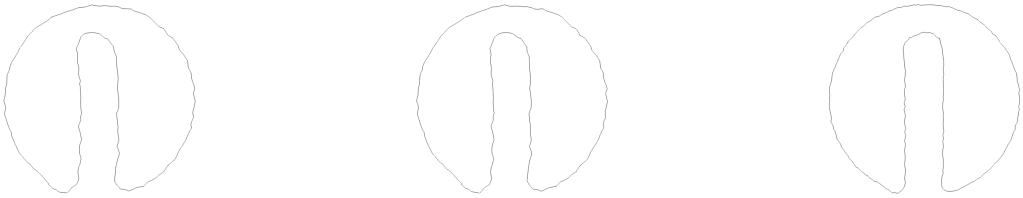
13.1 13.2

Figure 13: Initial shape in triangle meshes with different cell numbers:13.1 68805, 13.2 277605



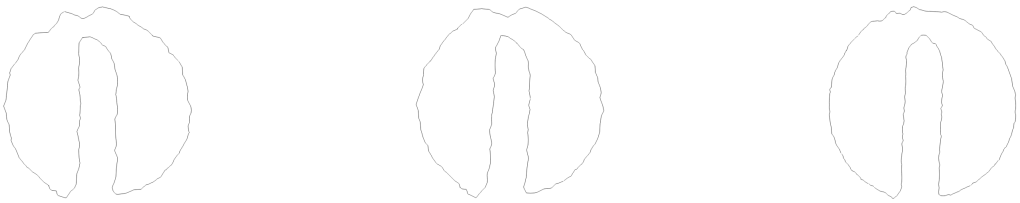
14.1 14.2 14.3

Figure 14: The shape after one circle with MULES method:14.1 68805 and $Co = 0.1$, 14.2 68805 and $Co = 0.5$,14.3 203965 and $Co = 0.5$



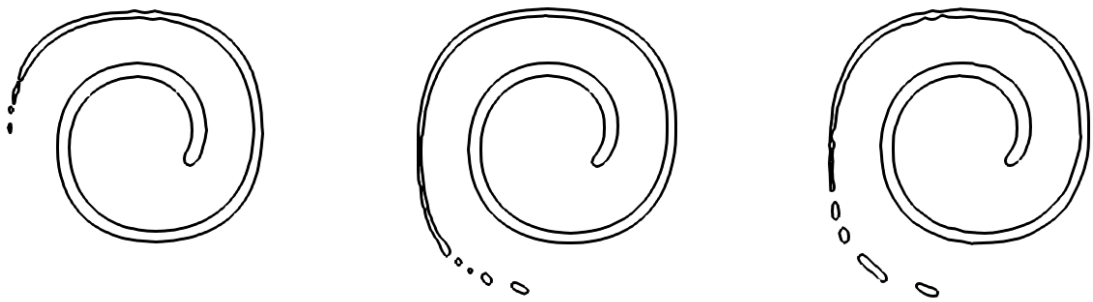
15.1 15.2 15.3

Figure 15: The shape after one circle with isoAdvector method:15.1 68805 and $Co = 0.1$, 15.2 68805 and $Co = 0.5$,15.3 203965 and $Co = 0.5$



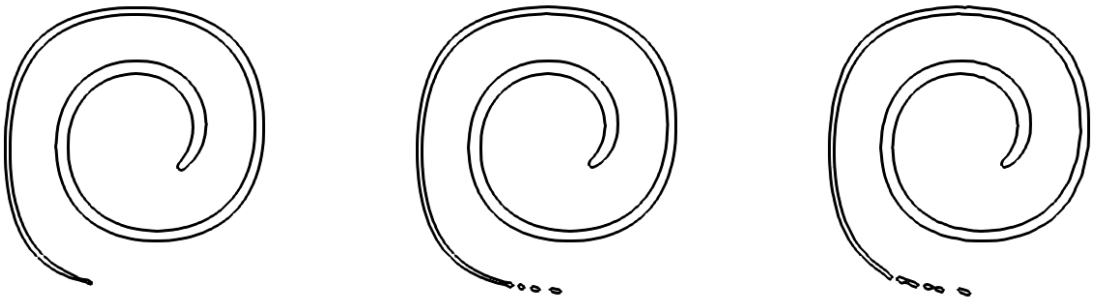
16.1 16.2 16.3

Figure 16: The shape after one circle with CLSAdvection method:16.1 68805 and $Co = 0.1$, 16.2 68805 and $Co = 0.5$,16.3 203965 and $Co = 0.5$



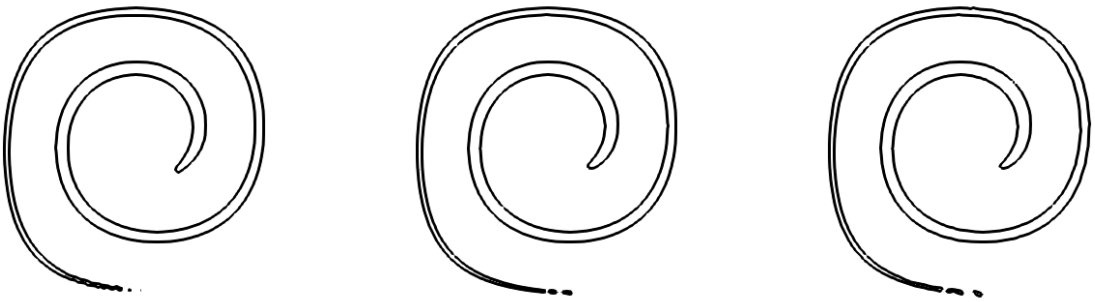
17.1 17.2 17.3

Figure 17: Spiraling discs at $T = 4$ with $Co = 0.5$.**??**: 100×100, MULES.**??**: 100×100, isoAdvector.**??**: 100×100.



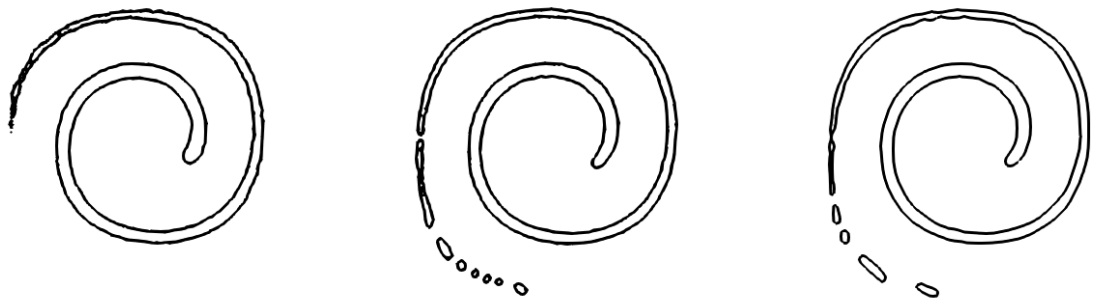
18.1 18.2 18.3

Figure 18: Spiraling discs at $T = 4$ with $Co = 0.5$.**??**: **??**: 200×200, MULES.**??**: 200×200, isoAdvector.**??**: 200×200, CLSAdvection.



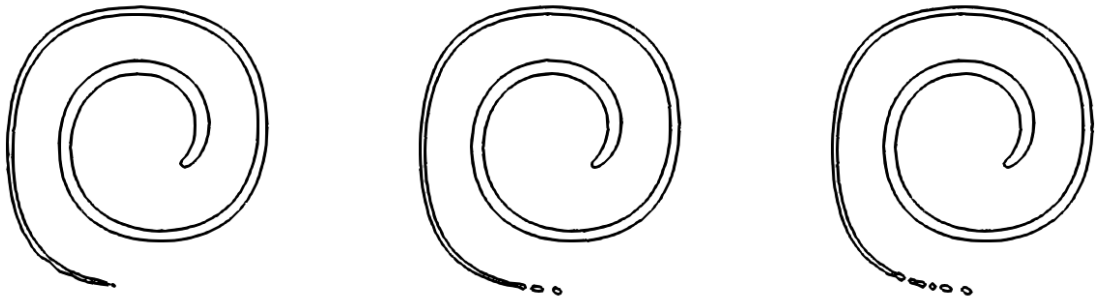
19.1 19.2 19.3

Figure 19: Spiraling discs at $T = 4$ with $Co = 0.5$.**??**:**??**: 300×300, MULES.**??**: 300×300, isoAdvector.**??**: 300×300, CLSAdvection.



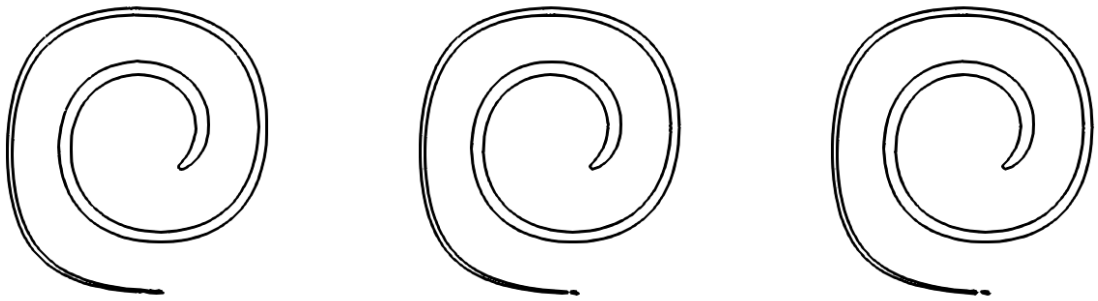
20.1 20.2 20.3

Figure 20: Spiraling discs at $T = 4$ with $Co = 0.5$.**??**: 100×100, MULES.**??**: 100×100, isoAdvector.**??**: 100×100, CLSAdvection.



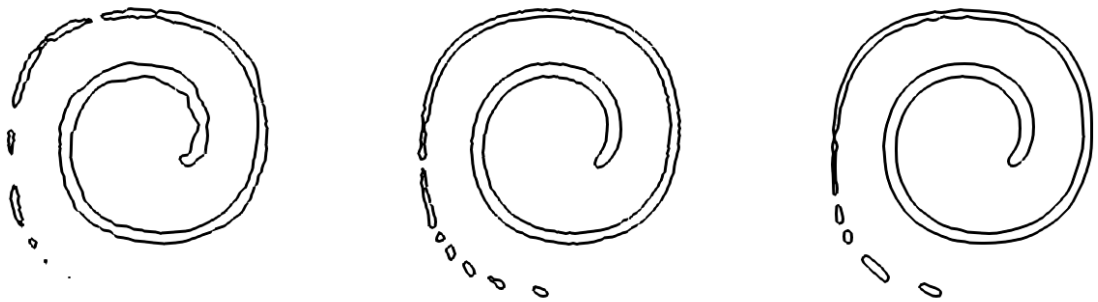
21.1 21.2 21.3

Figure 21: Spiraling discs at $T = 4$ with $Co = 0.5$.**??**: **??**: 200×200, MULES.**??**: 200×200, isoAdvector.**??**: 200×200, CLSAdvection.



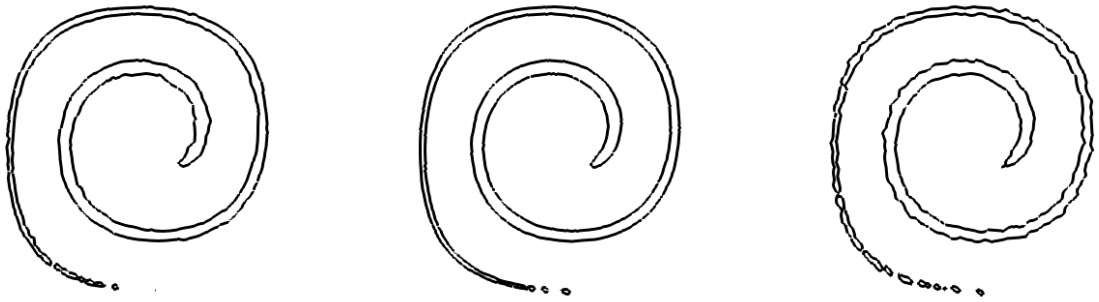
22.1 22.2 22.3

Figure 22: Spiraling discs at $T = 4$ with $Co = 0.5$.**??**:**??**: 300×300, MULES.**??**: 300×300, isoAdvector.**??**: *FinePoly*, CLSAdvection.



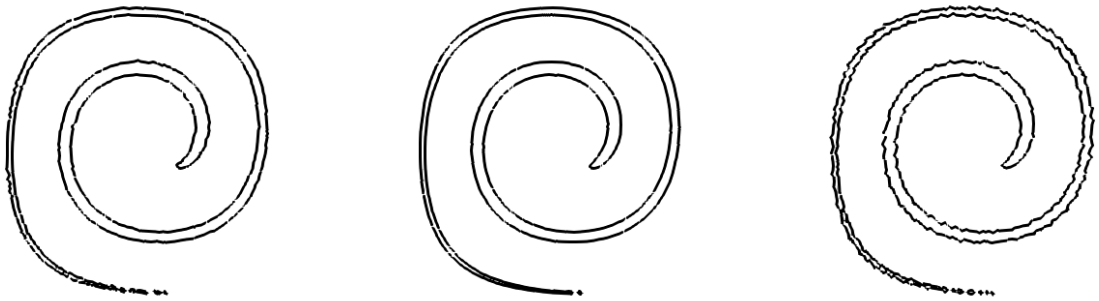
23.1 23.2 23.3

Figure 23: Spiraling discs at $T = 4$ with $Co = 0.5$.**??**: 100×100, MULES.**??**: 100×100, isoAdvector.**??**: 100×100, CLSAdvection.



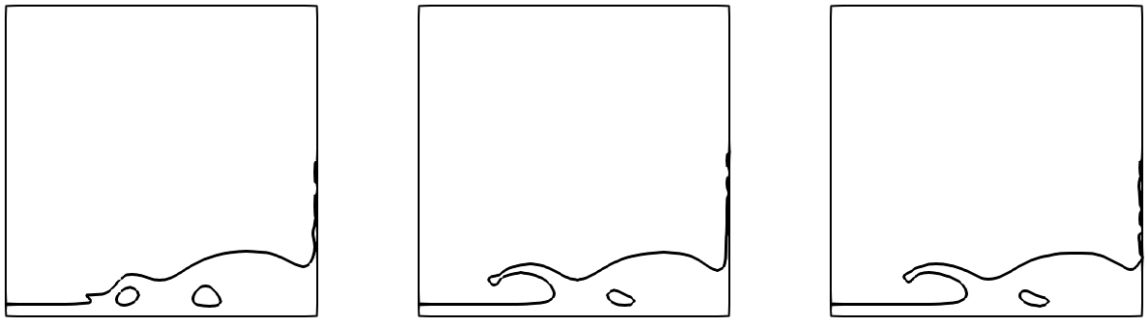
24.1 24.2 24.3

Figure 24: Spiraling discs at $T = 4$ with $Co = 0.5$.**??**: 200×200, MULES.**??**: 200×200, isoAdvector.**??**: 200×200, CLSAdvection.



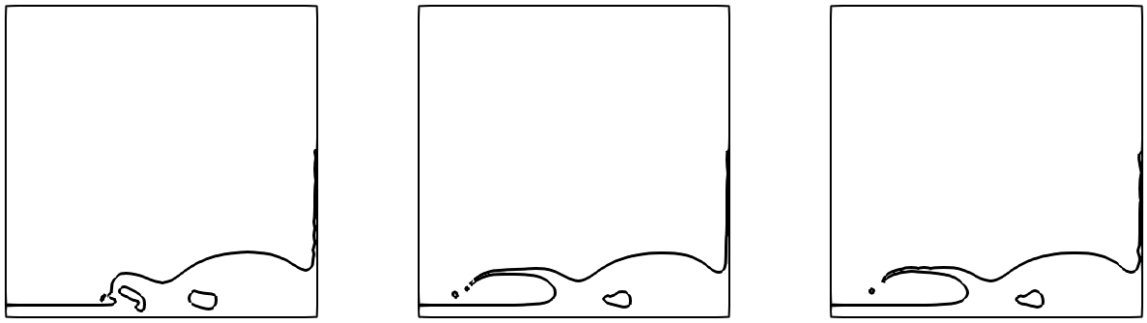
25.1 25.2 25.3

Figure 25: Spiraling discs at $T = 4$ with $Co = 0.5$.**??**: 300×300, MULES.**??**: 300×300, isoAdvector.**??**: $FinePoly$, CLSAdvection.



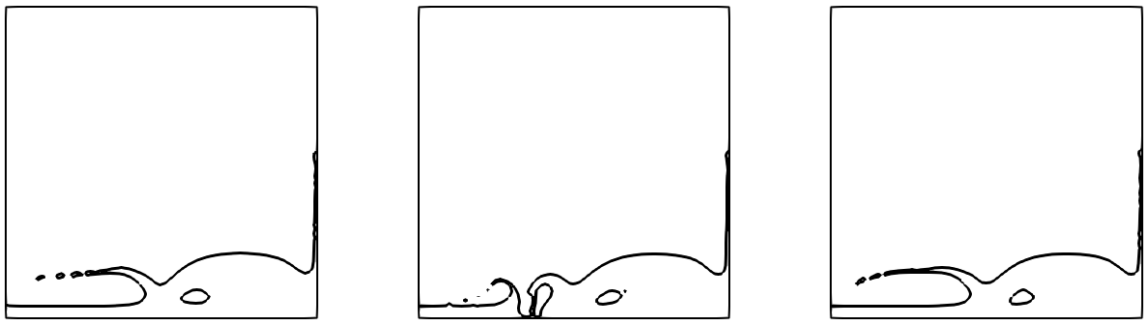
26.1 26.2 26.3

Figure 26: Check the mesh sensitivity of dam break case with different methods at time $T = 1s$.26.1: mesh size 100×100, MULES.26.2: mesh size 100×100, isoAdvector. 26.3: mesh size 100×100.



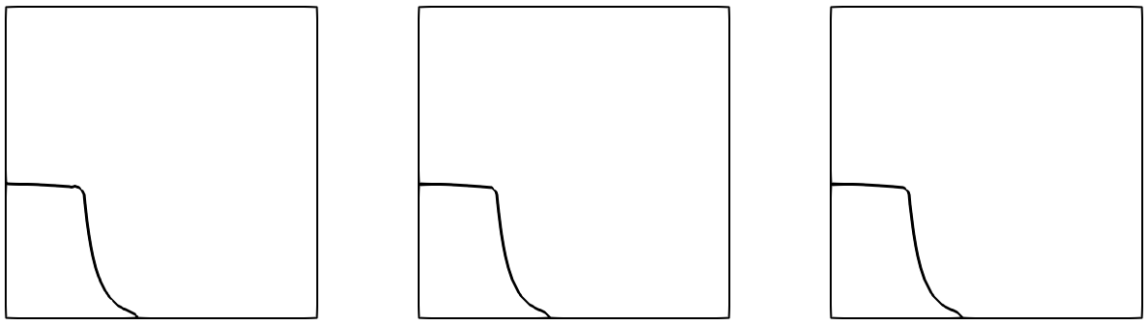
27.1 27.2 27.3

Figure 27: Check the mesh sensitivity of dam break case with different methods at time $T = 1s$. 27.1: mesh size 200×200, MULES.27.2: mesh size 200×200, isoAdvector. 27.3: mesh size 200×200, CLSAdvection.



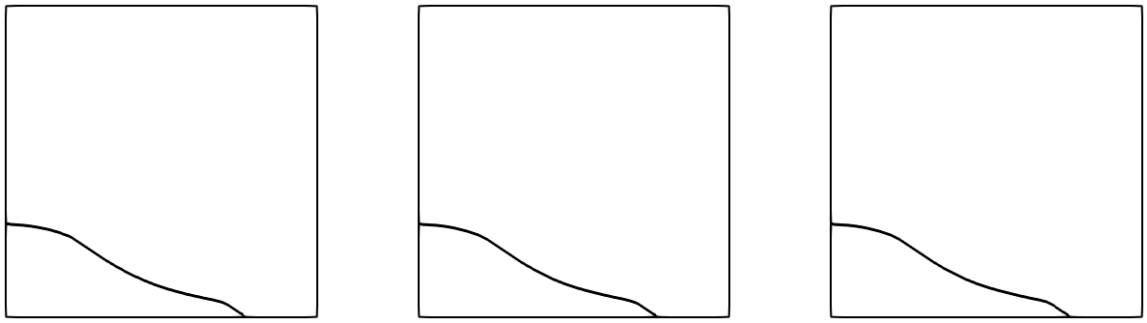
28.1 28.2 28.3

Figure 28: Check the mesh sensitivity of dam break case with different methods at time $T = 1s$.26.1:28.1: mesh size 300×300, MULES.28.2: mesh size 300×300, isoAdvector. 28.3: mesh size 300×300, CLSAdvection.



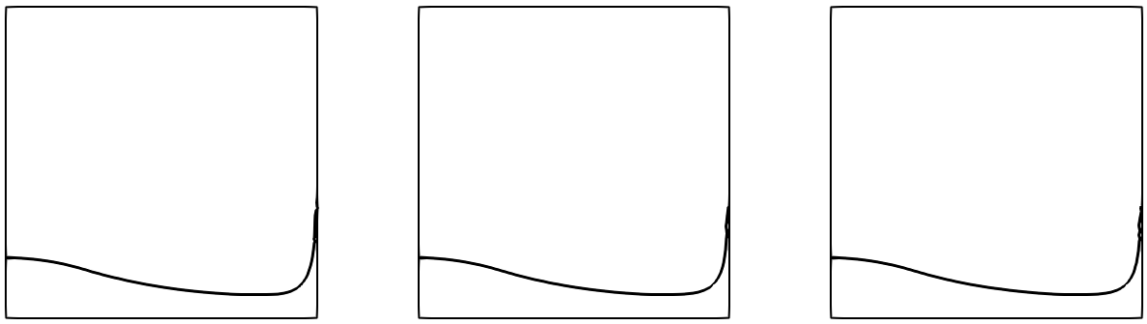
29.1 29.2 29.3

Figure 29: Dam break with different algorithm.



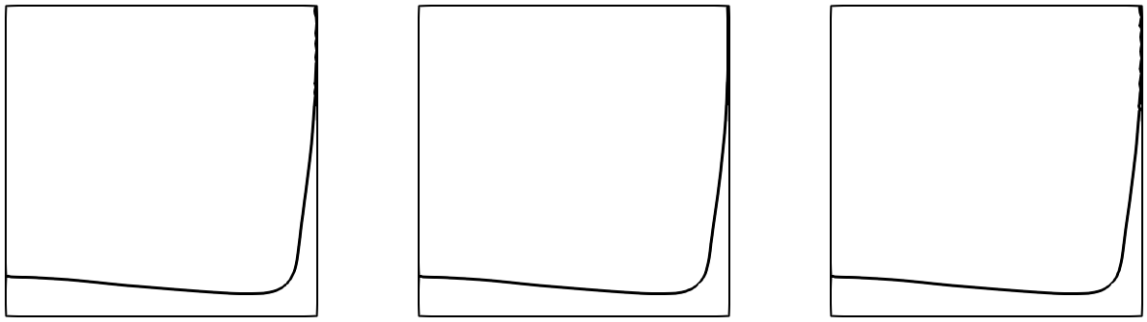
30.1 30.2 30.3

Figure 30: Dam break with different algorithm.



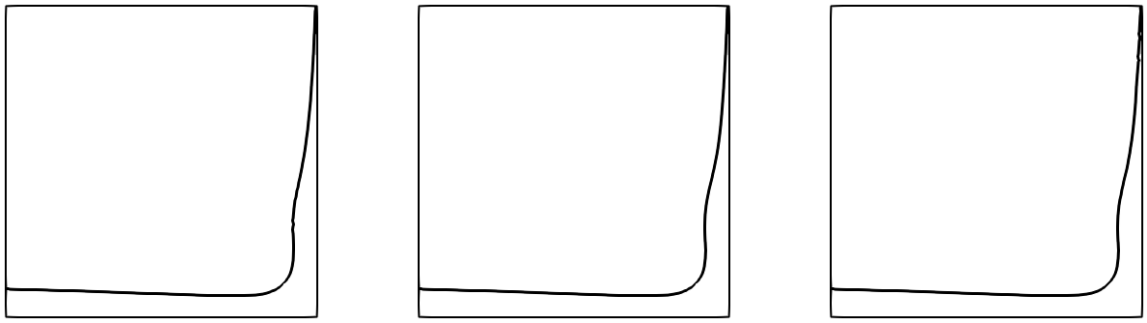
31.1 31.2 31.3

Figure 31: Dam break with different algorithm.



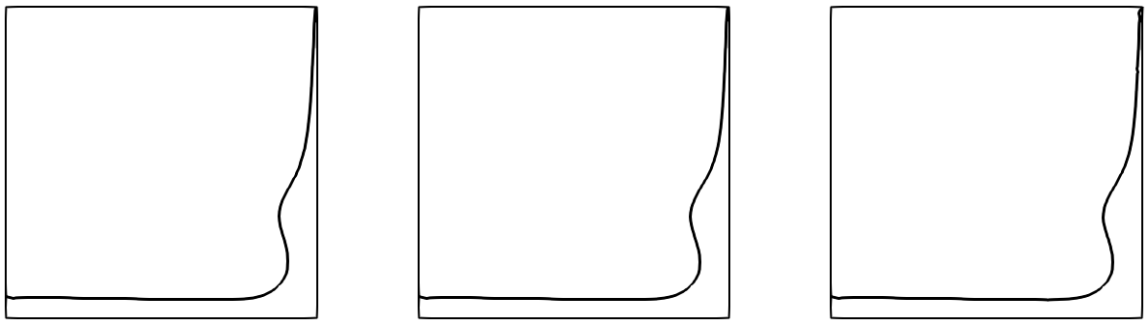
32.1 32.2 32.3

Figure 32: Dam break with different algorithm.



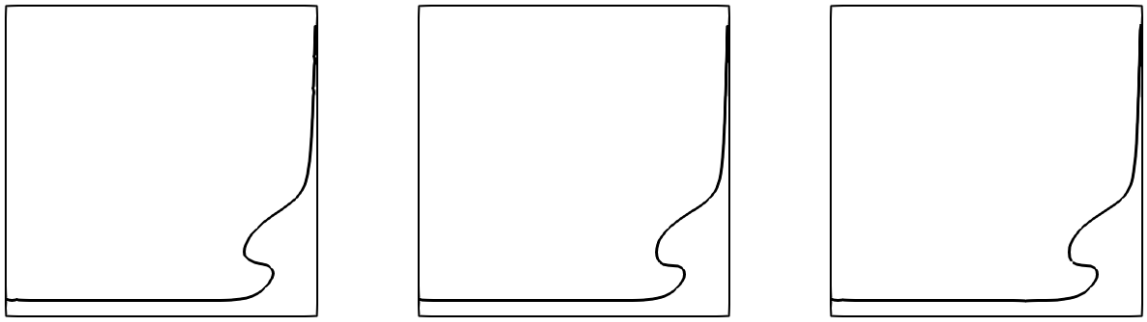
33.1 33.2 33.3

Figure 33: Dam break with different algorithm.



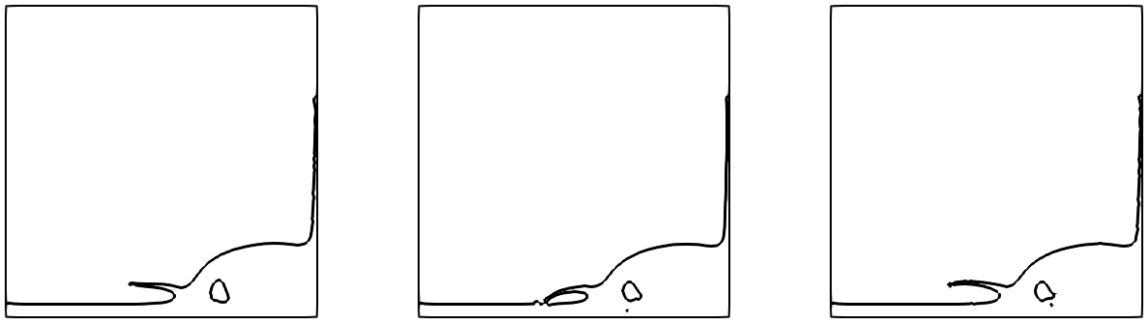
34.1 34.2 34.3

Figure 34: Dam break with different algorithm.



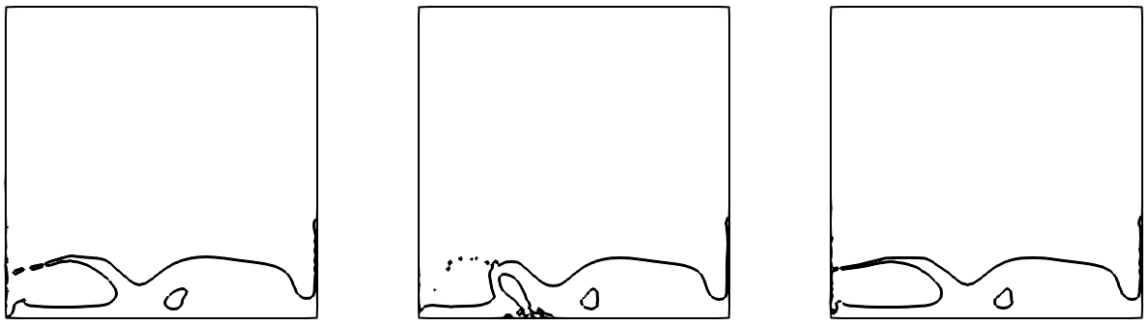
35.1 35.2 35.3

Figure 35: Dam break with different algorithm.



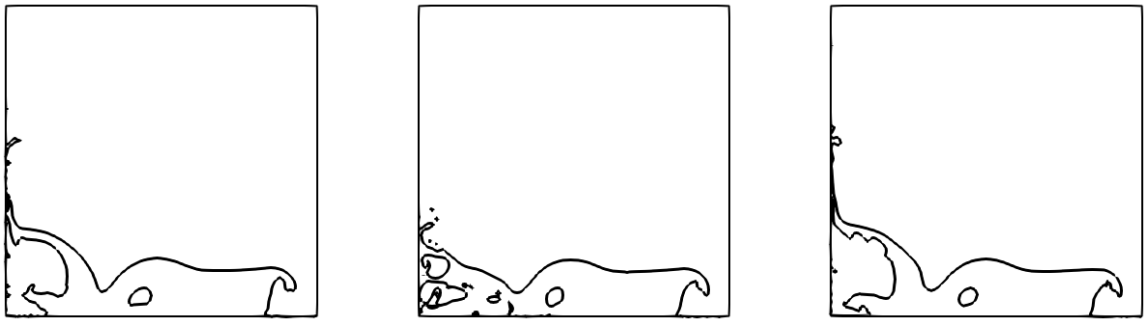
36.1 36.2 36.3

Figure 36: Dam break with different algorithm.



37.1 37.2 37.3

Figure 37: Dam break with different algorithm.



38.1 38.2 38.3

Figure 38: Dam break with different algorithm.

Table 1: Errors and calculation times for hex meshes of notched disc

	(N,Co)	MULES	isoAdvector	CLSAdvection
ε_V	(100,0.1)	1.06e-06	6.36e-11	7.57e-11
	(100,0.2)	4.79e-07	3.55e-10	1.65e-10
	(100,0.5)	5.67e-07	1.75e-07	4.10e-08
	(200,0.5)	4.88e-07	1.67e-06	6.92e-07
	(400,0.5)	3.33e-07	8.55e-07	3.97e-06
ε_M	(100,0.1)	1.00	1.00	0.99
	(100,0.2)	1.00	1.00	0.99
	(100,0.5)	1.00	1.00	0.99
	(200,0.5)	1.00	1.00	0.99
	(400,0.5)	1.00	1.00	1.00
ε_S	(100,0.1)	0.34	0.13	0.13
	(100,0.2)	0.34	0.13	0.13
	(100,0.5)	0.37	0.13	0.14
	(200,0.5)	0.22	0.061	0.065
	(400,0.5)	0.16	0.027	0.029
$\min(\alpha)$	(100,0.1)	-2.45e-36	0	0
	(100,0.2)	-8.03e-29	0	0
	(100,0.5)	-6.31e-20	0	0
	(200,0.5)	-4.40e-29	0	0
	(400,0.5)	-4.51e-47	0	0
$\max(\alpha) - 1$	(100,0.1)	-2.90e-06	0	0
	(100,0.2)	-1.29e-05	0	0
	(100,0.5)	-12.97e-04	0	0
	(200,0.5)	-2.49e-08	0	0
	(400,0.5)	-1.26e-10	0	0
T_{calc}	(100,0.1)	1160	417	1489
	(100,0.2)	648	390	873
	(100,0.5)	97	148	424
	(200,0.5)	2373	1461	2867
	(400,0.5)	7268	7214	12500

Table 2: Errors and calculation times for polygon meshes of notched disc

	(Cell Numbers,Co)	MULES	isoAdvector	CLSAdvection
ε_V	(51967,0.1)	9.30e-09	1.12e-10	1.14e-10
	(51967,0.5)	7.37e-07	4.05e-11	4.03e-11
	(203965,0.5)	2.77e-06	2.74e-11	2.94e-11
ε_M	(51967,0.1)	1.00	1.00	1.00
	(51967,0.5)	1.00	1.00	1.00
	(203965,0.5)	1.00	1.00	1.00
ε_S	(51967,0.1)	0.39	0.14	0.14
	(51967,0.5)	0.39	0.14	0.15
	(203965,0.5)	0.20	0.071	0.071
$\min(\alpha)$	(51967,0.1)	-8.22e-11	0	0
	(51967,0.5)	-2.54e-06	0	0
	(203965,0.5)	-7.72e-06	0	0
$\max(\alpha) - 1$	(51967,0.1)	-8.92e-06	0	0
	(51967,0.5)	-7.37e-05	0	0
	(203965,0.5)	1.44e-05	0	0
T_{calc}	(51967,0.1)	3170	803	1850
	(51967,0.5)	289	511	368
	(203965,0.5)	2521	1512	3550

Table 3: Errors and calculation times for triangle meshes of notched disc

	(Cell Numbers, Co)	MULES	isoAdvector	CLSAdvection
	(68805,0.1)	2.58e-07	5.22e-11	1.53e-05
	(68805,0.5)	9.93e-07	7.39e-10	1.30e-05
ε_V	(277605,0.5)	2.24e-07	7.60e-09	6.85e-06
	(68805,0.1)	1.00	1.00	1.00
	(68805,0.5)	1.00	1.00	1.00
ε_M	(277605,0.5)	1.00	1.00	1.00
	(68805,0.1)	0.44	0.11	0.15
	(68805,0.5)	0.47	0.12	0.32
ε_S	(277605,0.5)	0.30	0.062	0.30
	(68805,0.1)	-2.97e-24	0	0
	(68805,0.5)	-1.03e-17	0	0
$\min(\alpha)$	(277605,0.5)	-3.76e-18	0	0
	(68805,0.1)	-1.36e-04	0	0
	(68805,0.5)	-4.41e-04	0	0
$\max(\alpha) - 1$	(277605,0.5)	-3.10e-06	0	0
	(68805,0.1)	5446	1904	7102
	(68805,0.5)	670	1055	1900
T_{calc}	(277605,0.5)	5632	3474	11100

Table 4: Errors and calculation times for structured meshes of spiraling disc

	(Mesh size)	MULES	isoAdvector	CLSAdvection
	(100×100)	2.58e-07	5.22e-11	1.53e-05
	(200×200)	9.93e-07	7.39e-10	1.30e-05
ε_V	(300×300)	2.24e-07	7.60e-09	6.85e-06
	(100×100)	2.58e-07	5.22e-11	1.53e-05
	(200×200)	9.93e-07	7.39e-10	1.30e-05
ε_M	(300×300)	2.24e-07	7.60e-09	6.85e-06
	(100×100)	2.58e-07	5.22e-11	1.53e-05
	(200×200)	9.93e-07	7.39e-10	1.30e-05
ε_S	(300×300)	2.24e-07	7.60e-09	6.85e-06
	(100×100)	2.58e-07	5.22e-11	1.53e-05
	(200×200)	9.93e-07	7.39e-10	1.30e-05
$\min(\alpha)$	(300×300)	2.24e-07	7.60e-09	6.85e-06
	(100×100)	2.58e-07	5.22e-11	1.53e-05
	(200×200)	9.93e-07	7.39e-10	1.30e-05
$\max(\alpha) - 1$	(300×300)	2.24e-07	7.60e-09	6.85e-06
	(100×100)	2.58e-07	5.22e-11	1.53e-05
	(200×200)	9.93e-07	7.39e-10	1.30e-05
T_{calc}	(300×300)	2.24e-07	7.60e-09	6.85e-06

Table 5: Errors and calculation times for polygon meshes of spiraling disc

	(Mesh size)	MULES	isoAdvector	CLSAdvection
	(68805,0.1)	2.58e-07	5.22e-11	1.53e-05
	(68805,0.5)	9.93e-07	7.39e-10	1.30e-05
ε_V	(277605,0.5)	2.24e-07	7.60e-09	6.85e-06
	(68805,0.1)	1.00	1.00	1.00
	(68805,0.5)	1.00	1.00	1.00
ε_M	(277605,0.5)	1.00	1.00	1.00
	(68805,0.1)	0.44	0.11	0.15
	(68805,0.5)	0.47	0.12	0.32
ε_S	(277605,0.5)	0.30	0.062	0.30
	(68805,0.1)	-2.97e-24	0	0
	(68805,0.5)	-1.03e-17	0	0
$\min(\alpha)$	(277605,0.5)	-3.76e-18	0	0
	(68805,0.1)	-1.36e-04	0	0
	(68805,0.5)	-4.41e-04	0	0
$\max(\alpha) - 1$	(277605,0.5)	-3.10e-06	0	0
	(68805,0.1)	5446	1904	7102
	(68805,0.5)	670	1055	1900
T_{calc}	(277605,0.5)	5632	3474	11100

Table 6: Errors and calculation times for triangle meshes of spiraling disc

	(Mesh size)	MULES	isoAdvector	CLSAdvection
	(68805,0.1)	2.58e-07	5.22e-11	1.53e-05
	(68805,0.5)	9.93e-07	7.39e-10	1.30e-05
ε_V	(277605,0.5)	2.24e-07	7.60e-09	6.85e-06
	(68805,0.1)	1.00	1.00	1.00
	(68805,0.5)	1.00	1.00	1.00
ε_M	(277605,0.5)	1.00	1.00	1.00
	(68805,0.1)	0.44	0.11	0.15
	(68805,0.5)	0.47	0.12	0.32
ε_S	(277605,0.5)	0.30	0.062	0.30
	(68805,0.1)	-2.97e-24	0	0
	(68805,0.5)	-1.03e-17	0	0
$\min(\alpha)$	(277605,0.5)	-3.76e-18	0	0
	(68805,0.1)	-1.36e-04	0	0
	(68805,0.5)	-4.41e-04	0	0
$\max(\alpha) - 1$	(277605,0.5)	-3.10e-06	0	0
	(68805,0.1)	5446	1904	7102
	(68805,0.5)	670	1055	1900
T_{calc}	(277605,0.5)	5632	3474	11100

[7] M. Rudman, Volume-tracking methods for interfacial flow calculations, International journal for numerical methods in fluids 24 (1997) 671–691.

[8] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, Journal of Computational physics 114 (1994) 146–159.

[9] Y.-C. Chang, T. Hou, B. Merriman, S. Osher, A level set formulation of eulerian interface capturing methods for incompressible fluid flows, Journal of computational Physics 124 (1996) 449–464.

[10] S. Diot, M. M. François, An interface reconstruction method based on an analytical formula for 3d arbitrary convex cells, Journal of Computational Physics 305 (2016) 63–74.

[11] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, S. Zaleski, Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows, Journal of Computational Physics 152 (1999) 423–456.

[12] W. F. Noh, P. Woodward, Slic (simple line interface calculation), in: A. I. van de Vooren, P. J. Zandbergen (Eds.), Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28 – July 2, 1976 Twente University, Enschede, Springer Berlin Heidelberg, Berlin, Heidelberg, 1976, pp. 330–340.

[13] W. J. Rider, D. B. Kothe, Reconstructing volume tracking, Journal of Computational Physics 141 (1998) 112 – 152.

[14] D. L. Youngs, Time-Dependent Multi-material Flow with Large Fluid Distortion, 1982.

[15] J. Roenby, H. Bredmose, H. Jasak, A computational method for sharp interface advection, Royal Society open science 3 (2016) 160405.

[16] E. G. Puckett, A. S. Almgren, J. B. Bell, D. L. Marcus, W. J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, Journal of computational physics 130 (1997) 269–282.

[17] S. Muzaferija, Computation of free surface flows using interface-tracking and interface-capturing methods, Nonlinear water-wave interaction. Computational Mechanics, Southampton (1998).

[18] O. Ubbink, Numerical prediction of two fluid systems with sharp interfaces, Ph.D. thesis, Imperial College of Science, Technology and Medicine, 1997.

[19] H. G. Weller, A new approach to vof-based interface capturing methods for incompressible and compressible flow, OpenCFD Ltd., Report TR/HGW 4 (2008).

[20] H. Rusche, Computational fluid dynamics of dispersed two-phase flows at high phase fractions, Ph.D. thesis, Imperial College London (University of London), 2003.

[21] S. S. Deshpande, L. Anumolu, M. F. Trujillo, Evaluating the performance of the two-phase flow solver interfoam, Computational science & discovery 5 (2012) 014016.

[22] J. Roenby, isoadvector, 2016. `www.github.com/isoadvector`.

[23] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations, Journal of computational physics 79 (1988) 12–49.

[24] M. Sussman, E. Fatemi, S. Osher, P. Smereka, A level set approach for computing solutions to incompressible two- phase flow ii, in: 6 International Symposium on Computational Fluid Dynamics, 1995.

[25] M. Sussman, E. Fatemi, P. Smereka, S. Osher, An improved level set method for incompressible two-phase flows, Computers & Fluids 27 (1998) 663–680.

[26] S. Osher, R. Fedkiw, Level set methods and dynamic implicit surfaces, volume 153, Springer Science & Business Media, 2006.

[27] J. A. Sethian, Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, volume 3, Cambridge university press, 1999.

[28] D. L. Chopp, Computing minimal surfaces via level set curvature flow, Journal of computational physics 106 (1991) 77 – 91.

[29] F. Liu, Y. Xu, Y. Li, A coupled level-set and volume-of-fluid method for simulating axi-symmetric incompressible two-phase flows, Applied Mathematics and Computation 293 (2017) 112–130.

[30] G. Son, N. Hur, A coupled level set and volume-of-fluid method for the buoyancy-driven motion of fluid particles, Numerical Heat Transfer: Part B: Fundamentals 42 (2002) 523–542.

[31] T. Martin, I. Shevchuk, Implementation and validation of semi-implicit weno schemes using openfoam®, Computation 6 (2018) 6.

[32] Z. Ge, J.-C. Loiseau, O. Tammisola, L. Brandt, An efficient mass-preserving interface-correction level set/ghost fluid method for droplet suspensions under depletion forces, Journal of Computational Physics 353 (2018) 435–459.

[33] N. Johansson, Implementation of a standard level set method for incompressible two-phase flow simulations, 2011.

[34] F. Losasso, R. Fedkiw, S. Osher, Spatially adaptive techniques for level set methods and incompressible flow, Computers & Fluids 35 (2006) 995–1010.

[35] K. Luo, C. Shao, Y. Yang, J. Fan, A mass conserving level set method for detailed numerical simulation of liquid atomization, Journal of Computational Physics 298 (2015) 495–519.

[36] R. Nourgaliev, S. Wiri, N. Dinh, T. Theofanous, On improving mass conservation of level set by reducing spatial discretization errors, International journal of multiphase flow 31 (2005) 1329–1336.

[37] A. Salih, S. G. Moulic, Some numerical studies of interface advection properties of level set method, Sadhana 34 (2009) 271–298.

[38] W. H. Reed, T. Hill, Triangular mesh methods for the neutron transport equation, Technical Report, Los Alamos Scientific Lab., N. Mex.(USA), 1973.

[39] P. Rasetarinera, M. Hussaini, An efficient implicit discontinuous spectral galerkin method, Journal of Computational Physics 172 (2001) 718–738.

[40] J.-F. Remacle, N. Chevaugeon, E. Marchandise, C. Geuzaine, Efficient visualization of high-order finite elements, International Journal for Numerical Methods in Engineering 69 (2007) 750–771.

[41] D. Enright, F. Losasso, R. Fedkiw, A fast and accurate semi-lagrangian particle level set method, Computers & structures 83 (2005) 479–490.

[42] J. Strain, Semi-lagrangian methods for level set equations, Journal of Computational Physics 151 (1999) 498–533.

[43] D. Xiu, G. E. Karniadakis, A semi-lagrangian high-order method for navier–stokes equations, Journal of computational physics 172 (2001) 658–684.

[44] D. Adalsteinsson, J. A. Sethian, The fast construction of extension velocities in level set methods, Journal of Computational Physics 148 (1999) 2–22.

[45] D. L. Chopp, Another look at velocity extensions in the level set method, SIAM Journal on Scientific Computing 31 (2009) 3255–3273.

[46] A. Ovsyannikov, V. Sabelnikov, M. Gorokhovski, A new level set equation and its numerical assessments, in: Proceedings of the Summer Program, 2012, p. 315.

[47] V. Sabelnikov, A. Y. Ovsyannikov, M. Gorokhovski, Modified level set equation and its numerical assessment, Journal of Computational Physics 278 (2014) 1–30.

[48] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, Journal of Computational Physics 210 (2005) 225 – 246.

[49] E. Olsson, G. Kreiss, S. Zahedi, A conservative level set method for two phase flow ii, Journal of Computational Physics 225 (2007) 785 – 807.

[50] R. Chiodi, O. Desjardins, A reformulation of the conservative level set reinitialization equation for accurate and robust simulation of complex multiphase flows, Journal of Computational Physics 343 (2017) 186–200.

[51] P.-H. Chiu, Y.-T. Lin, A conservative phase field method for solving incompressible two-phase flows, Journal of Computational Physics 230 (2011) 185–204.

[52] Y. Sato, B. Ničeno, A conservative local interface sharpening scheme for the constrained interpolation profile method, International Journal for Numerical Methods in Fluids 70 (2012) 441–467.

[53] T. W. Sheu, C. Yu, P.-H. Chiu, Development of a dispersively accurate conservative level set scheme for capturing interface in two-phase flows, Journal of Computational Physics 228 (2009) 661–686.

[54] T. W. Sheu, C. Yu, P. Chiu, Development of level set method with good area preservation to predict interface in two-phase flows, International Journal for Numerical Methods in Fluids 67 (2011) 109–134.

[55] T. Nonomura, K. Kitamura, K. Fujii, A simple interface sharpening technique with a hyperbolic tangent function applied to compressible two-fluid modeling, Journal of Computational Physics 258 (2014) 95 – 117.

[56] M. Owkes, O. Desjardins, A discontinuous galerkin conservative level set scheme for interface capturing in multiphase flows, Journal of Computational Physics 249 (2013) 275–302.

[57] F. Xiao, Y. Honma, T. Kono, A simple algebraic interface capturing scheme using hyperbolic tangent function, International Journal for Numerical Methods in Fluids 48 (2005) 1023–1040.

[58] C. Walker, B. Muller, A conservative level set method for sharp interface multiphase flow simulation, in: ECCOMAS CFD, 2010.

[59] O. Desjardins, V. Moureau, H. Pitsch, An accurate conservative level set/ghost fluid method for simulating turbulent atomization, Journal of Computational Physics 227 (2008) 8395–8416.

[60] H. Kohno, J.-C. Nave, A new method for the level set equation using a hierarchical-gradient truncation and remapping technique, Computer Physics Communications 184 (2013) 1547 – 1554.

[61] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, M. L. Welcome, An adaptive level set approach for incompressible two-phase flows, Journal of Computational Physics 148 (1999) 81–124.

[62] D. Fuster, A. Bagué, T. Boeck, L. Le Moyne, A. Leboissetier, S. Popinet, P. Ray, R. Scardovelli, S. Zaleski, Simulation of primary atomization with an octree adaptive mesh refinement and vof method, International Journal of Multiphase Flow 35 (2009) 550–565.

[63] M. Herrmann, A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids, Journal of computational physics 227 (2008) 2674–2706.

[64] O. Desjardins, H. Pitsch, A spectrally refined interface approach for simulating multiphase flows, Journal of computational physics 228 (2009) 1658–1677.

[65] H. Kim, M.-S. Liou, Accurate adaptive level set method and sharpening technique for three dimensional deforming interfaces, Computers & Fluids 44 (2011) 111–129.

[66] D. Q. Nguyen, R. P. Fedkiw, M. Kang, A boundary condition capturing method for incompressible flame discontinuities, Journal of Computational Physics 172 (2001) 71–98.

[67] G. Son, A level set method for incompressible two-fluid flows with immersed solid boundaries, Numerical Heat Transfer, Part B: Fundamentals 47 (2005) 473–489.

[68] F. Gibou, L. Chen, D. Nguyen, S. Banerjee, A level set based sharp interface method for the multiphase incompressible navierstokes equations with phase change, Journal of Computational Physics 222 (2007) 536 – 555.

[69] S. Tanguy, A. Berlemont, Application of a level set method for simulation of droplet collisions, International journal of multiphase flow 31 (2005) 1015–1035.

[70] R. P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method), Journal of computational physics 152 (1999) 457–492.

[71] X.-D. Liu, R. P. Fedkiw, M. Kang, A boundary condition capturing method for poisson's equation on irregular domains, Journal of computational Physics 160 (2000) 151–178.

[72] M. Sussman, E. Fatemi, An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow, SIAM Journal on scientific computing 20 (1999) 1165–1191.

[73] G. Russo, P. Smereka, A remark on computing distance functions, Journal of Computational Physics 163 (2000) 51 – 67.

[74] D. Hartmann, M. Meinke, W. Schrder, Differential equation based constrained reinitialization for level set methods, Journal of Computational Physics 227 (2008) 6821 – 6845.

[75] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A pde-based fast local level set method, Journal of computational physics 155 (1999) 410–438.

[76] J. O. McCaslin, O. Desjardins, A localized re-initialization equation for the conservative level set method, Journal of Computational Physics 262 (2014) 408 – 426.

[77] D. Adalsteinsson, J. A. Sethian, A fast level set method for propagating interfaces, Journal of computational physics 118 (1995) 269–277.

[78] J. A. Sethian, A fast marching level set method for monotonically advancing fronts, Proceedings of the National Academy of Sciences 93 (1996) 1591–1595.

[79] J. A. Sethian, Evolution, implementation, and application of level set and fast marching methods for advancing fronts, Journal of computational physics 169 (2001) 503–555.

[80] D. Salac, The augmented fast marching method for level set reinitialization, arXiv preprint arXiv:1111.6903 (2011).

[81] R. F. Ausas, E. A. Dari, G. C. Buscaglia, A geometric mass-preserving redistancing scheme for the level set function, International journal for numerical methods in fluids 65 (2011) 989–1010.

[82] A. Salih, S. G. Moulic, A mass conservation scheme for level set method applied to multiphase incompressible flows, International Journal for Computational Methods in Engineering Science and Mechanics 14 (2013) 271–289.

[83] S.-R. Hysing, S. Turek, The eikonal equation: numerical efficiency vs. algorithmic complexity on quadrilateral grids, in: Proceedings of ALGORITMY, volume 22, 2005.

[84] M. Sussman, E. G. Puckett, A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows, Journal of Computational Physics 162 (2000) 301 – 337.

[85] M. Sussman, A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles, Journal of Computational Physics 187 (2003) 110 – 136.

[86] Z. Wang, J. Yang, F. Stern, A new volume-of-fluid method with a constructed distance function on general structured grids, Journal of Computational Physics 231 (2012) 3703–3722.

[87] Z. Wang, A. Y. Tong, A sharp surface tension modeling method for two-phase incompressible interfacial flows, International Journal for Numerical Methods in Fluids 64 (2010) 709–732.

[88] X. Lv, Q. Zou, Y. Zhao, D. Reeve, A novel coupled level set and volume of fluid method for sharp interface capturing on 3d tetrahedral grids, Journal of Computational Physics 229 (2010) 2573–2604.

[89] T. Ménard, S. Tanguy, A. Berlemont, Coupling level set/vof/ghost fluid methods: Validation and application to 3d simulation of the primary break-up of a liquid jet, International Journal of Multiphase Flow 33 (2007) 510–524.

[90] X. Yang, A. J. James, J. Lowengrub, X. Zheng, V. Cristini, An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids, Journal of Computational Physics 217 (2006) 364–394.

[91] G. Son, Efficient implementation of a coupled level-set and volume-of-fluid method for three-dimensional incompressible two-phase flows, Numerical Heat Transfer: Part B: Fundamentals 43 (2003) 549–565.

[92] D. Sun, W. Tao, A coupled volume-of-fluid and level set (voset) method for computing incompressible two-phase flows, International Journal of Heat and Mass Transfer 53 (2010) 645–655.

[93] C. E. Kees, I. Akkerman, M. W. Farthing, Y. Bazilevs, A conservative level set method suitable for variable-order approximations and unstructured meshes, Journal of Computational Physics 230 (2011) 4536–4558.

[94] Y. Wang, S. Simakhina, M. Sussman, A hybrid level set-volume constraint method for incompressible two-phase flow, Journal of Computational Physics 231 (2012) 6438–6471.

[95] Z. Cao, D. Sun, J. Wei, B. Yu, A coupled volume-of-fluid and level set method based on multi-dimensional advection for unstructured triangular meshes, Chemical Engineering Science 176 (2018) 560–579.

[96] S. Van der Pijl, A. Segal, C. Vuik, P. Wesseling, Computing three-dimensional two-phase flows with a mass-conserving level set method, Computing and Visualization in Science 11 (2008) 221–235.

[97] S. Van der Pijl, A. Segal, C. Vuik, P. Wesseling, A mass-conserving level-set method for modelling of multi-phase flows, International journal for numerical methods in fluids 47 (2005) 339–361.

[98] V. Le Chenadec, H. Pitsch, A 3d unsplit forward/backward volume-of-fluid approach and coupling to the level set method, Journal of computational physics 233 (2013) 10–33.

[99] P. Trontin, S. Vincent, J.-L. Estivalezes, J.-P. Caltagirone, A subgrid computation of the curvature by a particle/level-set method. application to a front-tracking/ghost-fluid method for incompressible flows, Journal of Computational Physics 231 (2012) 6990–7010.

[100] S. E. Hieber, P. Koumoutsakos, A lagrangian particle level set method, Journal of Computational Physics 210 (2005) 342–367.

[101] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, Journal of Computational physics 183 (2002) 83–116.

[102] L. Zhang, Clsadvector, 2018. `https://github.com/zlf0030/coupledAdvection`.

[103] J. U. Brackbill, D. B. Kothe, C. Zemach, A continuum method for modeling surface tension, Journal of computational physics 100 (1992) 335–354.

[104] T. R. C. M. Pringuey, Large eddy simulation of primary liquid-sheet breakup, Ph.D. thesis, University of Cambridge, 2012.

[105] T. Pringuey, R. S. Cant, High order schemes on three-dimensional general polyhedral meshesapplication to the level set method, Communications in Computational Physics 12 (2012) 1–41.

[106] C. Bilger, M. Aboukhedr, K. Vogiatzaki, R. S. Cant, Evaluation of two-phase flow solvers using level set and volume of fluid methods, Journal of Computational Physics 345 (2017) 665–686.

[107] T. Pringuey, R. S. Cant, Robust conservative level set method for 3d mixed-element meshesapplication to les of primary liquid-sheet breakup, Communications in Computational Physics 16 (2014) 403–439.

[108] S. Osher, C.-W. Shu, High-order essentially nonoscillatory schemes for hamilton–jacobi equations, SIAM Journal on numerical analysis 28 (1991) 907–922.

[109] T. Wang, H. Li, Y. Feng, D. Shi, A coupled volume-of-fluid and level set (voset) method on dynamically adaptive quadtree grids, International Journal of Heat and Mass Transfer 67 (2013) 70 – 73.

[110] V. R. Gopala, B. G. van Wachem, Volume of fluid methods for immiscible-fluid and free-surface flows, Chemical Engineering Journal 141 (2008) 204–221.

[111] S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, Journal of computational physics 31 (1979) 335–362.