

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/333600566>

A hybrid particle volume-of-fluid method for multiphase flows with surface tension

Preprint · June 2019

CITATIONS

0

READS

158

3 authors, including:



Petr Karnakov

ETH Zurich

11 PUBLICATIONS 16 CITATIONS

SEE PROFILE

A hybrid particle volume-of-fluid method for multiphase flows with surface tension

Petr Karnakov, Sergey Litvinov, Petros Koumoutsakos*

Computational Science and Engineering Laboratory, ETH Zürich, Clausiusstrasse 33, CH-8092 Zürich, Switzerland

Abstract

We present a particle method for estimating the curvature of interfaces and respective surface tension models in volume-of-fluid simulations of multiphase flows. Strings of particles are fitted, through an equilibration process, to the interface reconstruction from the volume-of-fluid discretization. The algorithm is conceptually simple and straightforward to implement as the same procedure is applied in every cell. We evaluate the hybrid method on a number of two- and three-dimensional benchmarks and illustrate its capabilities on simulations of turbulent multiphase flows and bubble coalescence. The results indicate that the curvature estimation is accurate even when the interface spans one cell per curvature radius and outperforms the generalized height-function method. The present method is in particular well suited for multiphase flow simulations with multiple interfaces discretized at low resolution.

Keywords: curvature, surface tension, volume-of-fluid, particles, coalescence

1. Introduction

Bubbles and drops are critical components of important industrial applications such as boiling and condensation [19], bubble column reactors [16], electrochemical cells [7] and physical systems involving air entrainment in plunging jets [20] and liquid jet atomization [22]. Simulations of such processes are challenged by the multiple scales of bubbles and their surface tension. Since the pioneering work of Brackbill *et al.* [9] in modeling surface tension with the Eulerian representation, a number of advances have been made [26], using level-sets [33] and volume-of-fluid methodologies (VOF) [29] to describe the interface and compute the surface tension.

The reconstruction of the interface in VOF methods is prone to inaccuracies that were shown to be eliminated in certain cases through a parabolic reconstruction of interfaces [27] and the balancing of pressure gradients with surface tension. The interface curvature estimation was further improved by the

method of height functions [10] that employs the discrete volume fraction field. The algorithm chooses a coordinate plane and integrates the volume fraction in columns perpendicular to the plane to obtain a function representing the distance from the interface to the plane. A well-defined height corresponds to a column crossing the interface exactly once such that its endpoints are on the opposite sides of the interface. The curvature is then estimated by finite differences on the plane which allows for high-order convergence [34]. However, the method requires that the heights are available on a sufficiently large stencil which imposes strong restrictions on the resolution: five cells per radius for circles and eight cells for spheres [25]. Modifications of the method aim to weaken this requirement by fitting an analytical function to the known values [27, 8, 11]. Heuristic criteria define whether the method of height functions or its modifications are applied in every cell. The first complete implementation of such approach was given by the generalized height-function method [25] which used parabolic fitting to heights from mixed directions and to centroids of the interface fragments. A similar approach was later implemented by [22].

*Corresponding author

Email address: petros@ethz.ch (Petros Koumoutsakos)

An alternative approach is the mesh-decoupled height function method [23] allowing for arbitrary orientation of the columns. Each height is computed from the intersection of the column and the fluid volume reconstructed by polyhedrons. However, in three dimensions the procedure involves complex and computationally expensive geometrical routines for triangulation of the shapes and still requires at least three cells per radius. The method of parabolic reconstruction directly from the volume fractions [12] has a high order convergence rate without restrictions on the minimal resolution in two dimensions. However, the extension of this algorithm to three dimensions is not straightforward.

We introduce a new method for computing the curvature in the volume-of-fluid framework which allows for solving transport problems with bubbles and drops at low resolution up to one cell per radius. The method relies on a reconstruction of the interface and it is applicable, but not limited, to the volume-of-fluid methods. The curvature estimation is obtained by fitting circular arcs to the reconstructed interface. A circular arc is represented by a string of particles. The fitting implies an evolution of the particles under constraints with forces attracting them to the interface.

We remark that the present approach is related to the concept of active contours [18]. The key differences include the imposition of hard constraints (particles belong to circular arcs) and the use of attraction forces based on the interface reconstruction. We also note that the present algorithm differs from the generalized height-function method [25, 22] as it is applied unconditionally in every cell. Furthermore, it is more accurate up to a resolution of eight cells per radius in three dimensions and even at a resolution of one cell per radius provides the relative curvature error below 0.1.

The paper is organized as follows: Section 2 describes the method for curvature estimation and the model of flows with surface tension as an application. Section 3 reports results on test cases involving spherical interfaces. Section 4 presents applications to turbulent flows and bubble coalescence. Section 5 concludes the study.

2. Numerical methods

This section describes a particle method for estimating the interface curvature from a discrete volume fraction field on two- and three-dimensional Cartesian meshes. The method is coupled with a volume-of-fluid (VOF) method for the simulation of incompressible multiphase flows with surface tension. We employ strings of particles to fit the interface using the reconstructed interface from the volume-of-fluid discretization. We distinguish the two- and three-dimensional implementation of the method and emphasize its utility for simulations at low resolutions.

2.1. VOF method for multiphase flows with surface tension

We incorporate the proposed curvature estimator to a numerical model describing two-component incompressible flows with surface tension. The system consists of the Navier-Stokes equations for the mixture velocity \mathbf{u} and pressure p

$$\nabla \cdot \mathbf{u} = 0, \quad (1)$$

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \nabla \cdot \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \mathbf{f}_\sigma + \rho \mathbf{g} \quad (2)$$

and the advection equation for the volume fraction α

$$\frac{\partial \alpha}{\partial t} + (\mathbf{u} \cdot \nabla) \alpha = 0 \quad (3)$$

with density $\rho = (1 - \alpha)\rho_1 + \alpha\rho_2$, dynamic viscosity $\mu = (1 - \alpha)\mu_1 + \alpha\mu_2$, gravitational acceleration \mathbf{g} and constant material parameters ρ_1, ρ_2, μ_1 and μ_2 . The surface tension force is defined as $\mathbf{f}_\sigma = \sigma \kappa \nabla \alpha$ with the surface tension coefficient σ and interface curvature κ .

We use a finite volume discretization based on the SIMPLE method for the pressure coupling [24, 14] and the second-order scheme QUICK [21] for convective fluxes. The advection equation is solved using the volume-of-fluid method PLIC with piecewise linear reconstruction [6] where the normals are computed using the mixed Youngs-centered scheme which is a combination of Youngs' scheme and the height functions. The cell-centered interface curvature κ is computed from the volume fraction using the proposed method. Our approximation of the

surface tension force is well-balanced [26] (i.e. the surface tension force is balanced by the pressure gradient if the curvature is uniform) and requires face-centered values of the curvature which are taken from the neighboring cell with minimal $|\alpha - \frac{1}{2}|$.

The algorithm is implemented on top of Cubism [36, 2], an open-source C++ framework for distributed parallel solvers on structured grids. To solve the linear systems, we use the GMRES method [28] for the momentum equation and the preconditioned conjugate gradient method [5] for the pressure correction implemented in the Hypre library [3, 13]. We also provide a visual web-based demonstration of the method for curvature estimation¹ and a reference implementation in Python².

2.2. Particles for curvature estimation in 2D

The interface curvature is estimated from a discrete volume fraction field on a two-dimensional Cartesian mesh. We fit circular arcs to a piecewise linear reconstruction of the interface. The arcs are represented as strings of particles subject to constraints and attracted to the interface through an equilibration process.

The reconstruction of the interface is built from the volume fraction field which ranges between 0 and 1 for pure components 1 and 2 respectively. Following [6], we compute the interface normals using the mixed Youngs-centered method on a 3×3 stencil. Then, the interface is reconstructed in each cell independently by a line segment cutting the cell into two parts according the estimated normal and the given volume fraction [30].

To estimate the curvature in a cell, we consider a set $L = \{L_l\}$ of line segments from a 5×5 stencil (the chosen stencils are justified in later sections). Each line segment $L_l = [\mathbf{a}_l, \mathbf{b}_l]$ has endpoints \mathbf{a}_l and \mathbf{b}_l , its unit normal \mathbf{n}_l points from component 2 to component 1 and the pair $(\mathbf{n}_l, \mathbf{b}_l - \mathbf{a}_l)$ is positively oriented. The curvature is computed by fitting a circular arc to the line segments. The circular arc is represented as a string of particles $\mathbf{x}_i \in \mathbb{R}^2$, $i = 1, \dots, N$ subject to constraints that prescribe the

distance h_p and require a uniform bending angle θ between the segments

$$|\mathbf{x}_{i+1} - \mathbf{x}_i| = h_p \quad i = 1, \dots, N-1, \quad (4)$$

$$(\mathbf{x}_{i+1} - \mathbf{x}_i) \times (\mathbf{x}_i - \mathbf{x}_{i-1}) = h_p^2 \sin \theta \quad i = 2, \dots, N-1. \quad (5)$$

Under these constraints the particles belong to a circular arc of curvature

$$\kappa(\theta) = \frac{\sqrt{2}}{h_p} \frac{\sin \theta}{\sqrt{1 + \cos \theta}}. \quad (6)$$

In addition to θ , we introduce the origin \mathbf{p} and the orientation angle ϕ and describe the positions of particles as

$$\mathbf{x}_i(\mathbf{p}, \phi, \theta) = \begin{cases} \mathbf{p} + \sum_{j=1}^{i-c} h_p \mathbf{e}(\phi + (j - \frac{1}{2})\theta) & i > c, \\ \mathbf{p} & i = c, \\ \mathbf{p} - \sum_{j=1}^{c-i} h_p \mathbf{e}(\phi - (j - \frac{1}{2})\theta) & i < c, \end{cases} \quad (7)$$

where $\mathbf{e}(\psi) = \cos \psi \mathbf{e}_x + \sin \psi \mathbf{e}_y$ and $c = (N-1)/2$ is the central particle (we require that N is odd). The parameters are illustrated in Figure 1a.

To define the force at position \mathbf{x} we find the nearest point $\mathbf{y} \in L_l$ on the interface and the corresponding line segment L_l . Then we find a factor δ such that

$$\mathbf{x}_L(\mathbf{x}, \kappa) = \mathbf{y} + \delta \mathbf{n}_l \quad (8)$$

belongs to a circular arc of curvature κ through the endpoints of L_l , where κ is an estimation of curvature. Such δ is given by

$$\delta = \sqrt{\frac{1}{\kappa^2} - d^2} - \sqrt{\frac{1}{\kappa^2} - w^2} = \frac{\kappa(w^2 - d^2)}{\sqrt{1 - \kappa^2 d^2} + \sqrt{1 - \kappa^2 w^2}}, \quad (9)$$

where $d = |\mathbf{y} - \mathbf{c}_l|$, $w = |\mathbf{a}_l - \mathbf{c}_l|$ and $\mathbf{c}_l = (\mathbf{a}_l + \mathbf{b}_l)/2$. Finally, the force is

$$\mathbf{f}(\mathbf{x}, \kappa) = \eta (\mathbf{x}_L(\mathbf{x}, \kappa) - \mathbf{x}), \quad (10)$$

where $\eta \in [0, 1]$ is a relaxation parameter. Figure 1b illustrates the force computation.

To find the equilibrium positions of particles we use an iterative algorithm. We introduce vector notation for particle positions and forces

$$\mathbf{X}(\mathbf{p}, \phi, \theta) = [\mathbf{x}_i(\mathbf{p}, \phi, \theta), i = 1, \dots, N], \quad (11)$$

$$\mathbf{F}(\mathbf{p}, \phi, \theta) = [\mathbf{f}(\mathbf{x}_i(\mathbf{p}, \phi, \theta), \kappa(\theta)), i = 1, \dots, N] \quad (12)$$

¹Visual demonstration: <https://cselab.github.io/hydro/grid.html>

²Reference implementation: <https://cselab.github.io/hydro/curv.py>

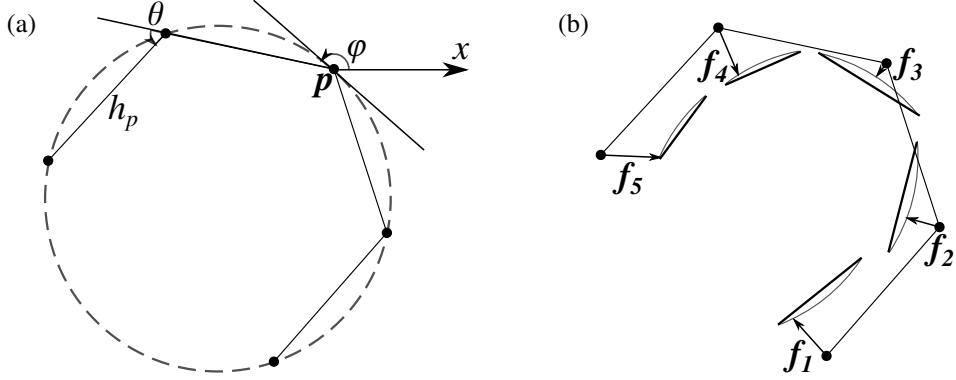


Figure 1: (a) All particles belong to a circle of curvature κ and their positions are defined by the central particle p , the orientation angle ϕ and the bending angle θ . (b) Line segments represent the interface and circular arcs of curvature κ pass through their endpoints. The force acts on each particle towards the circular arc from the nearest line segment.

and define the scalar product as

$$\mathbf{X} \cdot \mathbf{Y} = \sum_{i=1}^N \mathbf{X}_i \cdot \mathbf{Y}_i. \quad (13)$$

Initially the particles are arranged along the line segment L_l at which we estimate the curvature. The central particle is placed at the segment center such that $\mathbf{p}^0 = (\mathbf{a}_l + \mathbf{b}_l)/2$, the vectors $\mathbf{e}(\phi^0)$ and \mathbf{n}_l are orthogonal and positively oriented and the bending angle is zero $\theta^0 = 0$. Iteration m starts with computing the forces $\mathbf{F}^m = \mathbf{F}(\mathbf{p}^m, \phi^m, \theta^m)$ from positions $\mathbf{X}^m = X(\mathbf{p}^m, \phi^m, \theta^m)$. Without constraints, the correction of positions at one step would be

$$\Delta\mathbf{X} = \mathbf{F}. \quad (14)$$

With constraints, we aim to find the corrections of parameters $\Delta\mathbf{p}$, $\Delta\phi$ and $\Delta\theta$ giving the closest correction of positions in the least-square sense, i.e. by minimizing a functional

$$\|\mathbf{F} - \Delta\mathbf{X}\|_2 \rightarrow \min. \quad (15)$$

Every iteration consists of three steps each correcting one parameter: \mathbf{p} , ϕ and θ . The functional is approximately transformed in terms of the parameter correction. For instance, in terms of $\Delta\phi$ the functional

$$\|\mathbf{F} - \frac{\partial\mathbf{X}}{\partial\phi}(\mathbf{p}, \phi, \theta)\Delta\phi\|_2 \rightarrow \min \quad (16)$$

represents a quadratic function so the optimal correction is

$$\Delta\phi = \frac{\mathbf{F} \cdot \mathbf{D}_\phi}{\mathbf{D}_\phi \cdot \mathbf{D}_\phi}, \quad (17)$$

where $\mathbf{D}_\phi = \frac{\partial\mathbf{X}}{\partial\phi}(\mathbf{p}, \phi, \theta)$. The same procedure is applied to the bending angle θ . The origin p is corrected by the force acting on the central particle. We note that correcting the origin by the mean force instead, would lead to stronger spurious flows in the case of a static droplet described in Section 3.2. After each step, the correction $\Delta\mathbf{X}$ is subtracted from \mathbf{F} . This results in the following steps:

Step 1. Correct \mathbf{p} by the force acting on the central particle

$$\begin{aligned} \mathbf{p}^{m+1} &= \mathbf{p}^m + \mathbf{f}_c^m, \\ \mathbf{X}^* &= \mathbf{X}(\mathbf{p}^{m+1}, \phi^m, \theta^m), \\ \mathbf{F}^* &= \mathbf{F}^m - (\mathbf{X}^* - \mathbf{X}^m). \end{aligned}$$

Step 2. Correct ϕ

$$\begin{aligned} \phi^{m+1} &= \phi^m + \frac{\mathbf{F}^* \cdot \mathbf{D}_\phi}{\mathbf{D}_\phi \cdot \mathbf{D}_\phi}, \\ \mathbf{X}^{**} &= \mathbf{X}(\mathbf{p}^{m+1}, \phi^{m+1}, \theta^m), \\ \mathbf{F}^{**} &= \mathbf{F}^* - (\mathbf{X}^{**} - \mathbf{X}^*), \end{aligned}$$

where $\mathbf{D}_\phi = \frac{\partial\mathbf{X}}{\partial\phi}(\mathbf{p}^{m+1}, \phi^m, \theta^m)$.

Step 3. Correct θ

$$\begin{aligned} \theta^{m+1} &= \theta^m + \frac{\mathbf{F}^{**} \cdot \mathbf{D}_\theta}{\mathbf{D}_\theta \cdot \mathbf{D}_\theta}, \\ \mathbf{X}^{m+1} &= \mathbf{X}(\mathbf{p}^{m+1}, \phi^{m+1}, \theta^{m+1}), \end{aligned}$$

where $\mathbf{D}_\theta = \frac{\partial\mathbf{X}}{\partial\theta}(\mathbf{p}^{m+1}, \phi^{m+1}, \theta^m)$.

If the iterations have converged, the following equilibrium conditions are satisfied

$$\mathbf{f}_c = \mathbf{0}, \quad (18)$$

$$\mathbf{F} \cdot \frac{\partial \mathbf{X}}{\partial \phi} = 0, \quad (19)$$

$$\mathbf{F} \cdot \frac{\partial \mathbf{X}}{\partial \theta} = 0. \quad (20)$$

We stop the iterations once

$$E_m = \frac{\max_i |\mathbf{x}_i^m - \mathbf{x}_i^{m-1}|_\infty}{\eta h} < \varepsilon_p \quad (21)$$

or

$$m > m_{\max}, \quad (22)$$

where E_m is the maximum difference after iteration m , h is the mesh step, η is the relaxation parameter from (10), ε_p is the convergence tolerance and m_{\max} is the maximum number of iterations. The curvature is obtained from the bending angle θ at the last iteration by equation (6).

We note that the computation of forces can be simplified by setting $\delta = 0$ in (8) without a significant effect on the accuracy. However, the original formulation ensures that the method recovers the exact curvature if the endpoints of all line segments belong to a circle.

2.3. Particles for curvature estimation in 3D

In three dimensions, the interface is reconstructed following [6] as well, and the normal vectors are computed using the Youngs-centered method on a $3 \times 3 \times 3$ stencil. In this case the reconstructed interface is a set of planar convex polygons. We compute the mean curvature as the average over multiple cross sections and thus reduce the problem to the two-dimensional case. The algorithm to estimate the mean curvature in a cell consists of the following steps illustrated in Figure 2:

Step 1. Collect a set P of polygons from a $5 \times 5 \times 5$ stencil centered at the target cell. Determine the unit normal \mathbf{n} of the target polygon and the center \mathbf{x}_c as the mean over its vertices.

Step 2. Compute the curvature κ_j in each cross section $j = 0, \dots, N_s - 1$:

- Define a plane passing through \mathbf{x}_c and containing vectors \mathbf{n} and

$$\boldsymbol{\tau} = \cos(\pi j/N_s)\boldsymbol{\tau}_1 + \sin(\pi j/N_s)\boldsymbol{\tau}_2,$$

where $\boldsymbol{\tau}_1 = \mathbf{n} \times \mathbf{e}$, $\boldsymbol{\tau}_2 = \mathbf{n} \times \boldsymbol{\tau}_1$ and \mathbf{e} is one of the unit vectors $\{\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z\}$ providing the minimal $|\mathbf{n} \cdot \mathbf{e}|$.

- Intersect each polygon in P with the plane and collect non-empty intersections in a set of line segments $L = \{[\mathbf{a}_l, \mathbf{b}_l]\}$.
- Construct a set $\hat{L} = \{[\hat{\mathbf{a}}_l, \hat{\mathbf{b}}_l]\}$ from L by computing local two-dimensional coordinates of the endpoints

$$\hat{\mathbf{x}} = ((\mathbf{x} - \mathbf{x}_c) \cdot \boldsymbol{\tau}, (\mathbf{x} - \mathbf{x}_c) \cdot \mathbf{n}) \in \mathbb{R}^2,$$

where $\mathbf{x} = \mathbf{a}_l$ or $\mathbf{x} = \mathbf{b}_l$.

- Apply the procedure from Section 2.2 to \hat{L} to compute the curvature κ_j .

Step 3. Compute the mean curvature

$$\kappa = \frac{1}{N_s} \sum_{j=0}^{N_s-1} \kappa_j. \quad (23)$$

2.4. Convergence of iterations

We demonstrate the convergence of iterations for the equilibration of particles on the test case of estimating the curvature of a sphere introduced in Section 3.1. Figure 3 shows the maximum of E_m over all cells containing the interface at resolution $R/h = 4$. In all computations we set the relaxation parameter to $\eta = 0.5$ and use the convergence criteria with $\varepsilon_p = 10^{-5}$ and $m_{\max} = 20$.

2.5. Sensitivity to parameters

Estimation of the interface curvature in three dimensions depends on three parameters: the number of particles per string N , the number of cross sections N_s and the distance between the particles h_p . We express the distance between the particles as

$$h_p = \frac{H_p h}{N - 1}, \quad (24)$$

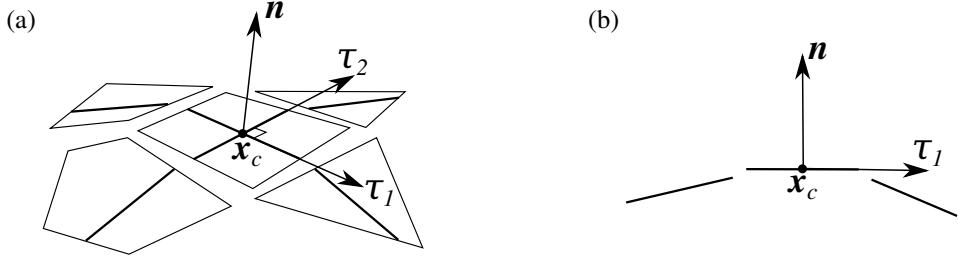


Figure 2: Curvature estimation in three dimensions in case $N_s = 2$. (a) Interface polygons with the cross sections. (b) One cross section in local two-dimensional coordinates.

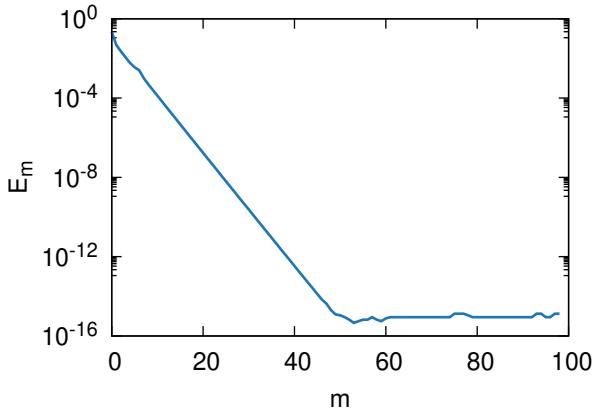


Figure 3: Convergence history for the maximum difference after one iteration over all particles on a sphere at resolution $R/h = 4$ and with $\eta = 0.5$.

where H_p defines the length of the string relative to the mesh step and, therefore, acts as a stencil size.

We set the parameters to $N = 9$, $N_s = 2$ and $H_p = 4$ and examine their influence by estimating the curvature of a sphere at various resolutions following the test case in Section 3.1. We vary each parameter independently. The figures present the median error over 100 samples for the center. As seen from Figure 4, the number of particles has a minor influence on the result. Nevertheless, we observe that $N = 3$ provides a two times larger error for bubbles at resolutions about one cell per radius. The influence of N_s in Figure 5 is also small which is expected for a sphere. However, more complex shapes such as those observed during the bubble coalescence in Section 4.2, require at least $N_s = 2$ as shown in Figure 7. Figure 6 shows a stronger influence of the string length H_p . Increasing the value from $H_p = 2$ to $H_p = 4$ reduces the error by a factor of ten.

3. Test cases

We examine the capabilities of the proposed methods on two and three-dimensional benchmark problems: curvature of a sphere, a static droplet and a translating droplet. The volume fraction is initialized by the exact volume cut by a sphere (circle) [32].

We compare the results with the generalized height-function method implemented in Basilisk [1, 25]. In Basilisk, the height function is evaluated in cells containing the interface and a particular technique is chosen for computing the curvature depending on the resolution, i.e. the number of available heights. If the heights are available on a stencil of 3×3 cells, the code uses second-order finite differences. If the resolution is insufficient, it fits a paraboloid to heights from mixed directions and to centroids of the interface in neighboring cells.

3.1. Curvature

The volume fraction field represents a single sphere (circle) of radius R . We vary the number of cells per radius R/h and consider 100 samples for the center from a uniform distribution over the octant (quadrant) of the cell, i.e. sampling each coordinate from the range $[0, h/2]$. We compute the relative curvature error in L_2 and L_∞ norms

$$L_2(\kappa) = \left(\frac{1}{|I|} \sum_{i \in I} \left(\frac{\kappa_i - \kappa_{\text{ex}}}{\kappa_{\text{ex}}} \right)^2 \right)^{1/2}, \quad (25)$$

$$L_\infty(\kappa) = \max_{i \in I} \left| \frac{\kappa_i - \kappa_{\text{ex}}}{\kappa_{\text{ex}}} \right|, \quad (26)$$

where I is the indices of cells containing the interface (i.e. cells i for which $0 < \alpha_i < 1$) and κ_{ex} is the exact curvature (i.e. $\kappa_{\text{ex}} = 2/R$ for sphere and $\kappa_{\text{ex}} = 1/R$ for circle).

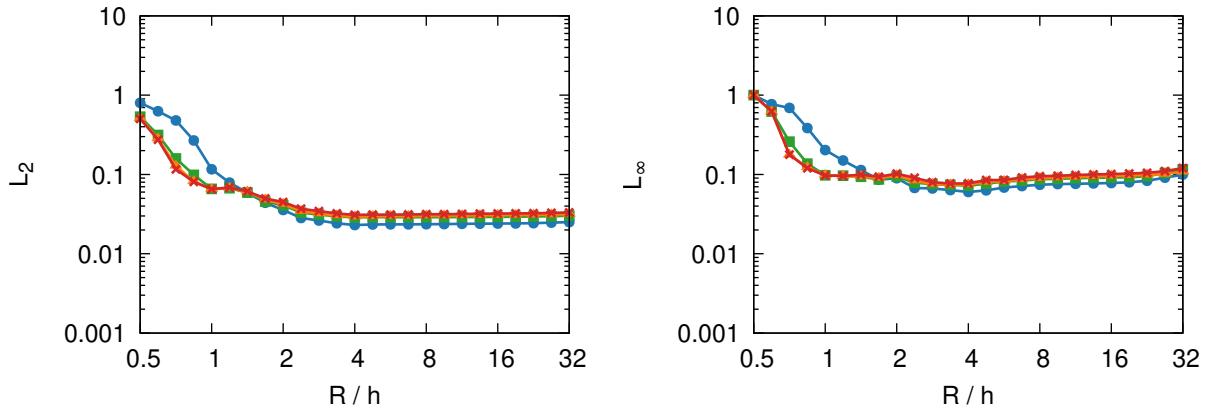


Figure 4: Curvature error for sphere in L_2 and L_∞ norms depending on the resolution for various values of the number of particles $N = 3$ —●—, 5 —■—, 7 —▲— and 9 —★—.

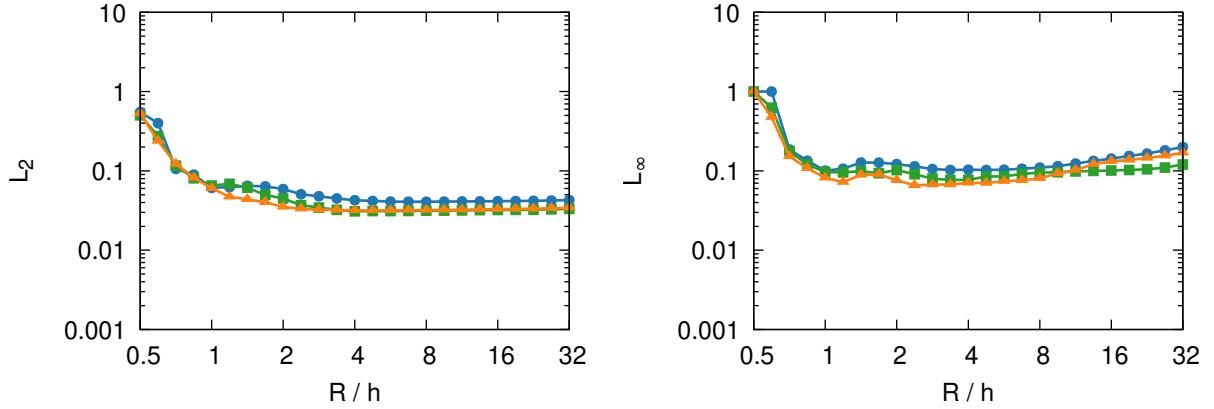


Figure 5: Curvature error for sphere in L_2 and L_∞ norms depending on the resolution for various values of number of cross sections $N_s = 1$ —●—, 2 —■— and 3 —▲—.

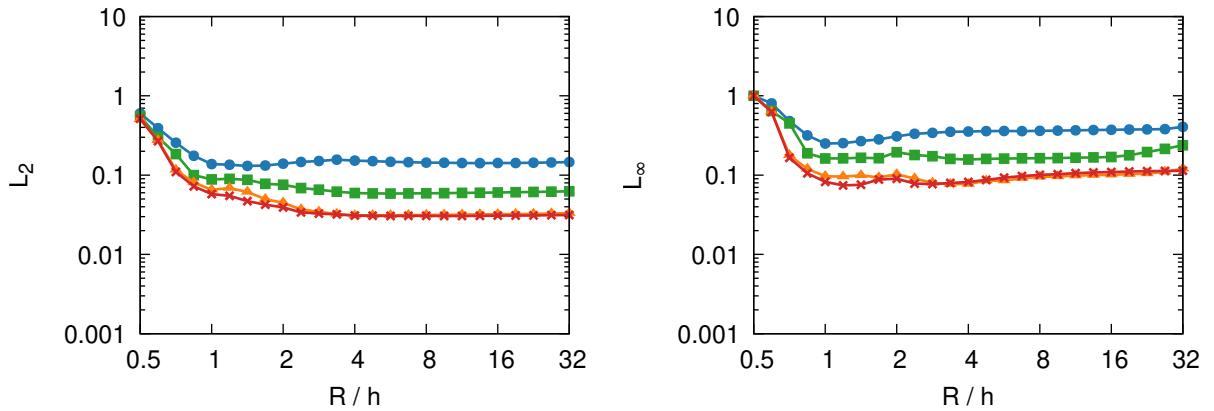


Figure 6: Curvature error for sphere in L_2 and L_∞ norms depending on the resolution for various values of the particle string length $H_p = 2$ —●—, 3 —■—, 4 —▲— and 5 —★—.

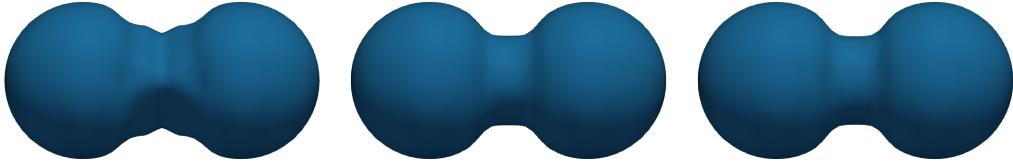


Figure 7: Isosurfaces of the volume fraction with $R/h = 9.6$ at $t/T = 0.17$ depending on the number of cross sections $N_s = 1$ (left), 2 (center) and 3 (right).

Figure 10 shows the error in comparison to Basilisk. At low resolutions, our method is more accurate in terms of the L_2 -error up to eight cells per radius in 3D (and four cells in 2D) and at resolutions below two cells per radius the error is a factor of ten smaller. Figures 8-9 show the final configuration of particles at resolutions below two cells per radius.

Unlike the present method, at high resolutions the method of height functions shows a second-order convergence of the relative error. For our method the relative error approaches a constant (corresponding to a first-order convergence of the absolute curvature). We note, however, that this behavior is sensitive to disturbances of the volume fraction field that may arise from applying an advection scheme.

3.2. Static droplet

We apply the model described in Section 2.1 for a spherical (circular) droplet in equilibrium. The initial velocity is zero and the volume fraction field represents a single sphere (circle) of radius R . We assume that both components have the same density and viscosity such that $\rho_1 = \rho_2 = \rho$ and $\mu_1 = \mu_2 = \mu$. This leaves one dimensionless parameter, the Laplace number characterizing the ratio between the surface tension and inertial forces

$$\text{La} = \frac{2\sigma\rho R}{\mu^2}, \quad (27)$$

which we set to $\text{La} = 1200$. We solve the problem on a mesh of 128^3 cells (or 128^2 cells in 2D) placing the droplet center in the corner and imposing the symmetry conditions on the adjacent boundaries. The other boundaries are free-slip walls. We vary the number of cells per radius R/h and advance the solution until time $T = \rho(2R)^2/\mu$.

In the exact solution, the velocity remains zero and the pressure experiences a jump at the interface given by the Laplace

pressure $p_L = \sigma\kappa_{\text{ex}}$. The numerical solutions develop spurious currents due to an imbalance between the pressure gradient and surface tension. We compute the maximum magnitude of the spurious velocity

$$U_{\max} = \max_{i \in C} |\mathbf{u}_i|, \quad (28)$$

the corresponding Weber number

$$\text{We}_{\max} = \frac{2\rho RU_{\max}^2}{\sigma} \quad (29)$$

and the pressure jump

$$\Delta p = \max_{i \in C} p_i - \min_{i \in C} p_i, \quad (30)$$

where C is the indices of all cells.

Figures 11-12 show the values of We_{\max} and the relative pressure jump $\Delta p/p_L$ at $t = T$ depending on the resolution. The evolution of We_{\max} for two selected resolutions is shown in Figure 13. We observe that the solutions from Basilisk converge with time to a zero spurious flow at fine resolution starting from $R/h = 9.51$ cells in 3D (and 3.36 cells in 2D). This demonstrates the existence of a volume fraction field for which the method of height functions provides a uniform curvature field. However, the convergence rate significantly reduces for $R/h = 5.66$, 11.31, and 22.63. Furthermore, the equilibrium shape is symmetric and, therefore, incompatible with advection. The spurious flow occurs in a more realistic scenario such as the translating droplet case discussed in Section 3.3.

Meanwhile, our method provides a lower magnitude of the spurious flow at low resolutions below $R/h = 8$ cells in 3D (and 2.83 cells in 2D) and a better accuracy for the pressure jump while in Basilisk it drops to zero.

3.3. Translating droplet

We extend the previous case by adding a uniform initial velocity field $\mathbf{u}(\mathbf{x}, 0) = \mathbf{U} = U\mathbf{l}/|\mathbf{l}|$ where $\mathbf{l} = (1, 0.8, 0.6)$ in 3D

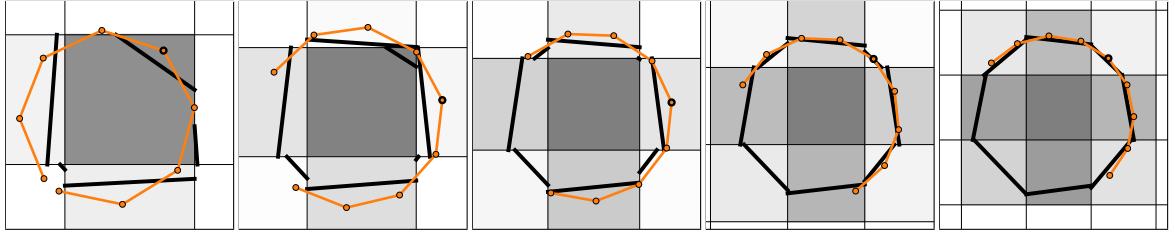


Figure 8: Line segments of the interface (black) and positions of particles (orange) from one selected cell for a circle at resolutions $R/h = 0.59, 0.71, 0.84, 1.0$ and 1.19 . The central particle is highlighted by a thicker edge.

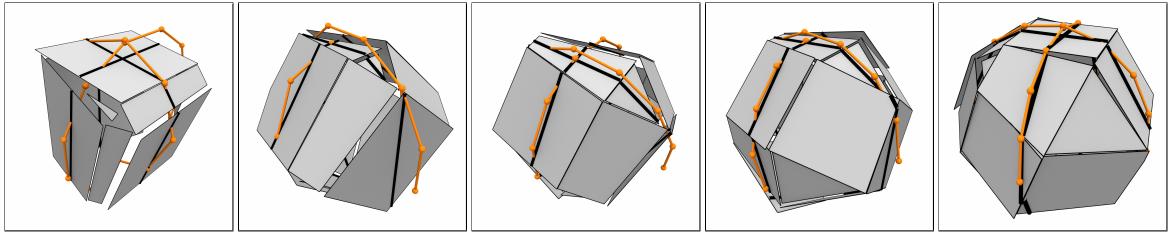


Figure 9: Polygons of the interface, cross sections (black) and positions of particles (orange) from one selected cell for a sphere at resolutions $R/h = 0.59, 0.71, 0.84, 1.0$ and 1.19 .

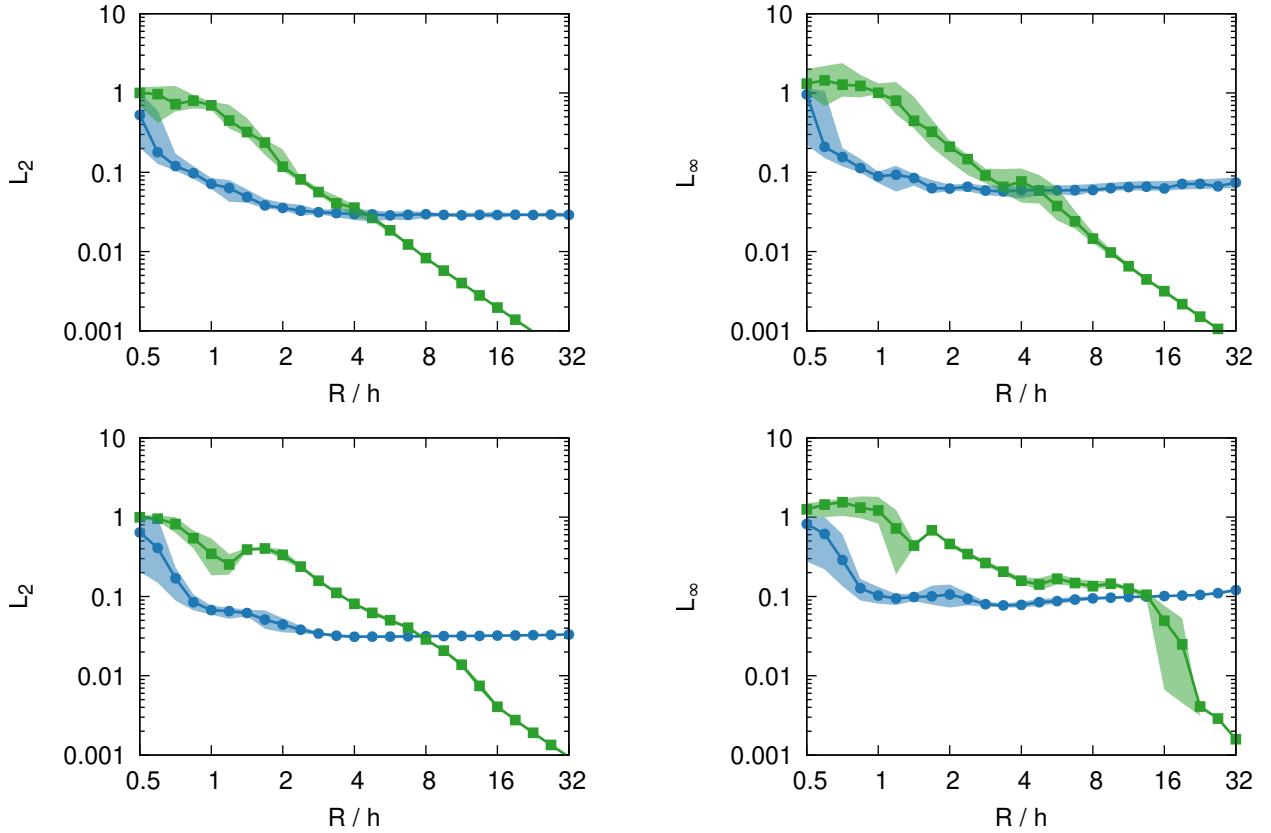


Figure 10: Curvature error for a circle (top) and sphere (bottom) in L_2 and L_∞ norms depending on the resolution: present —●— and Basilisk —■—. The lines show the median and the shades show the 10% and 90% percentiles for random positions of the center.

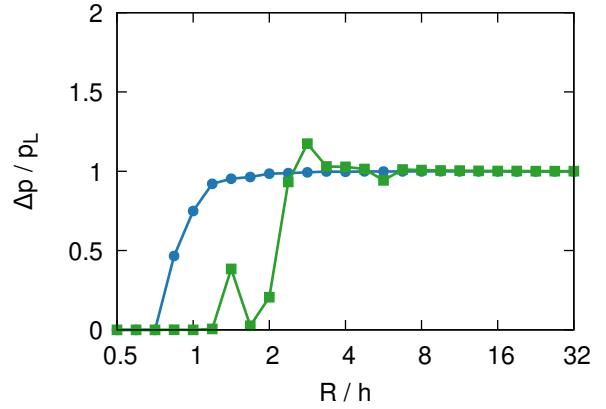
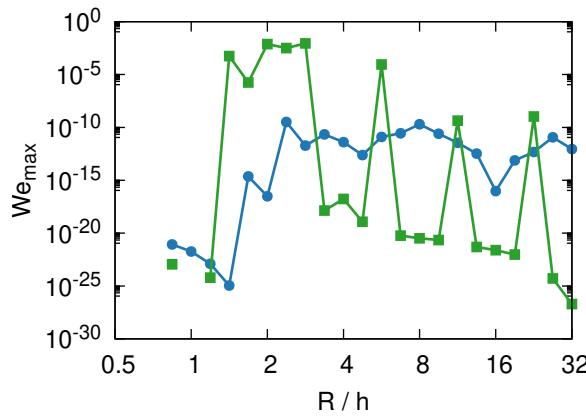


Figure 11: Velocity error and the pressure jump for a static droplet in 2D depending on the resolution: present —●— and Basilisk —■—.

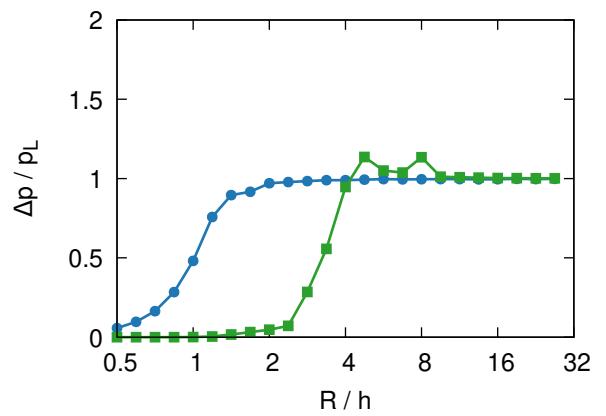
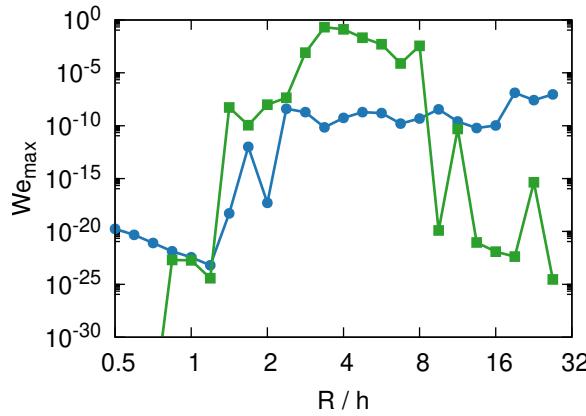


Figure 12: Velocity error and the pressure jump for a static droplet in 3D depending on the resolution: present —●— and Basilisk —■—.

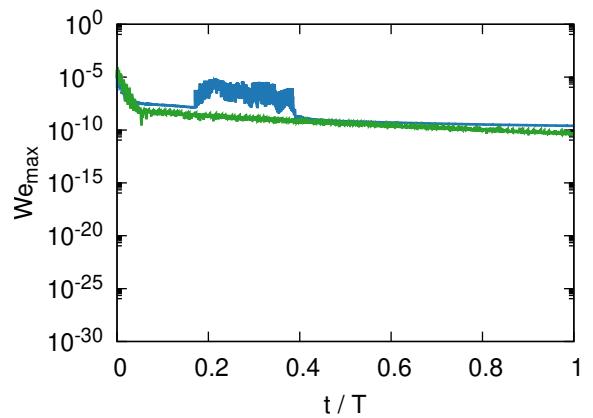
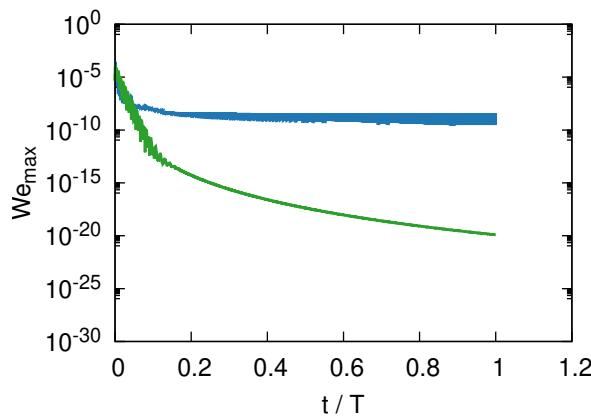


Figure 13: Evolution of the velocity error for a static droplet in 3D at resolutions $R/h = 9.51$ (left) and 11.31 (right): present —●— and Basilisk —■—.

and $I = (1, 0.8)$ in 2D. The additional parameter of the problem is the Weber number

$$\text{We} = \frac{2\rho RU^2}{\sigma}, \quad (31)$$

which we set to $\text{We} = 0.1$ while keeping $\text{La} = 1200$. We solve the problem in a periodic domain on a mesh of 128^3 cells (or 128^2 cells in 2D). We vary the number of cells per radius R/h and advance the solution until time $T = 2R/U$.

The magnitude of the spurious flow is computed relative to the initial velocity as the maximum over all cells

$$U_{\max} = \max_{i \in C} |\mathbf{u}_i - \mathbf{U}| \quad (32)$$

and definitions of We_{\max} and Δp follow (29) and (30). Figures 14-15 show We_{\max} and the relative pressure jump $\Delta p/p_L$ depending on the resolution. The quantities are averaged over $t \in [T/2, T]$ and the evolution of We_{\max} for two selected resolutions is shown in Figure 16. Our method provides lower magnitudes of the spurious flow than Basilisk and more accurate values of the pressure jump at resolutions below $R/h = 9.51$ cells in 3D (and $R/h = 5.66$ cells in 2D). Furthermore, it shows a first-order convergence for We_{\max} throughout all resolutions (except for a decrease in 3D above 16 cells per radius).

4. Applications

4.1. Taylor-Green vortex with bubble

The Taylor-Green vortex is a classical benchmark for the capabilities of flow solvers to simulate single-phase turbulent flows [35]. Here we extend the formulation by adding a gaseous phase. The problem is solved in a periodic domain $[0, 2\pi]^3$ with the initial velocity

$$u_x = \sin x \cos y \cos z, \quad (33)$$

$$u_y = -\cos x \sin y \cos z, \quad (34)$$

$$u_z = 0 \quad (35)$$

and a single bubble of radius $R = 0.2$ placed at $(2, 2, 2)$. Parameters of the problem are the Reynolds number $\text{Re} = \rho_1/\mu_1$ and the Weber number $\text{We} = 2\rho_1 R/\sigma$. Here we choose $\text{Re} = 800$

and $\text{We} = 4$. The density and viscosity ratios are set to $\rho_2/\rho_1 = 0.01$ and $\mu_2/\mu_1 = 0.01$.

Figure 17 shows the trajectory of the bubble at various resolutions in comparison to Basilisk. Both methods converge to the same solution with the mesh refinement. However, the accuracy of our method at $R/h = 6.11$ is comparable to that of the other method at $R/h = 12.23$. Furthermore, at lower resolutions, Basilisk produces solutions with a bubble breakup. One example of such behavior is presented in Figure 18 with $R/h = 6.11$. Regions of higher vorticity magnitude (shown in red) near the bubble indicate the spurious flow which later leads to the breakup. Our method maintains the spherical shape at all resolutions.

To evaluate the computational cost of the curvature estimation, we perform the same test on a mesh of 256^3 cells with 890 bubbles of smaller radius $R = 0.1$ seeded uniformly in the domain. The gas volume fraction amounts to 2% and the interface is distributed over 1.6% of the cells. Running the test on Piz Daint supercomputer using 86 compute nodes takes about 5 hours and 20% of the time is spent on the curvature estimation. We note that our current implementation can be further optimized.

4.2. Coalescence of bubbles

Coalescence of bubbles and drops is an actively studied phenomenon commonly found in nature and industry [4, 31]. Here we consider coalescence of two spherical bubbles. The problem is solved in a periodic domain $[0, 1]^3$ with zero initial velocity and two tangent bubbles of radius $R = 0.15$ placed along the x -axis. The only parameter of the problem is the Ohnesorge number

$$\text{Oh} = \frac{\mu_1}{\sqrt{\rho_1 R \sigma}}, \quad (36)$$

which we set to $\text{Oh} = 0.007$. The density and viscosity ratios are set to $\rho_2/\rho_1 = 0.01$ and $\mu_2/\mu_1 = 0.01$.

We refer to [31] for a detailed experimental study of bubble coalescence. The process starts with the formation of a neck connecting the bubbles which then propagates along the bubble surface. Figures 19-20 show the isosurfaces of the volume

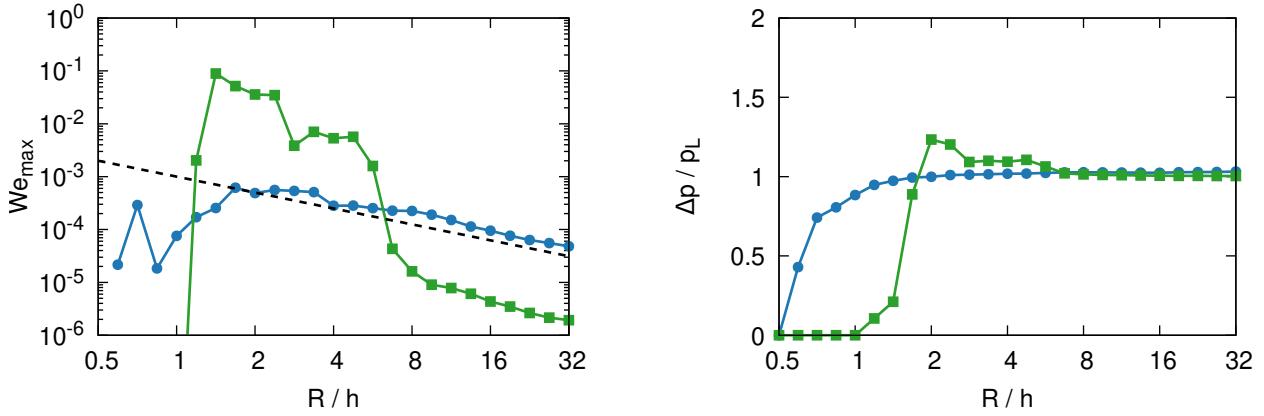


Figure 14: Velocity error and the pressure jump for a translating droplet in 2D depending on the resolution: present —●—, Basilisk —■— and first-order convergence - - -.

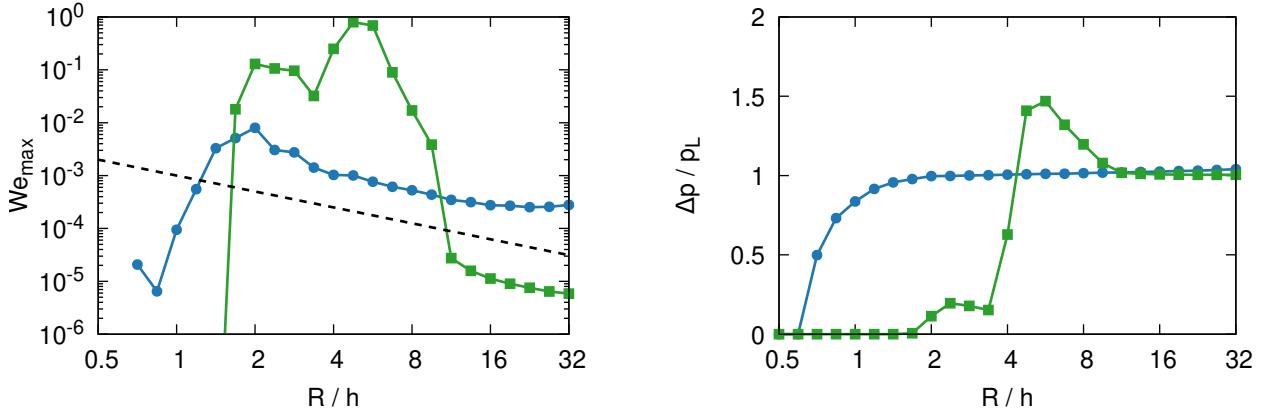


Figure 15: Velocity error and the pressure jump for a translating droplet in 3D depending on the resolution: present —●—, Basilisk —■— and first-order convergence - - -.

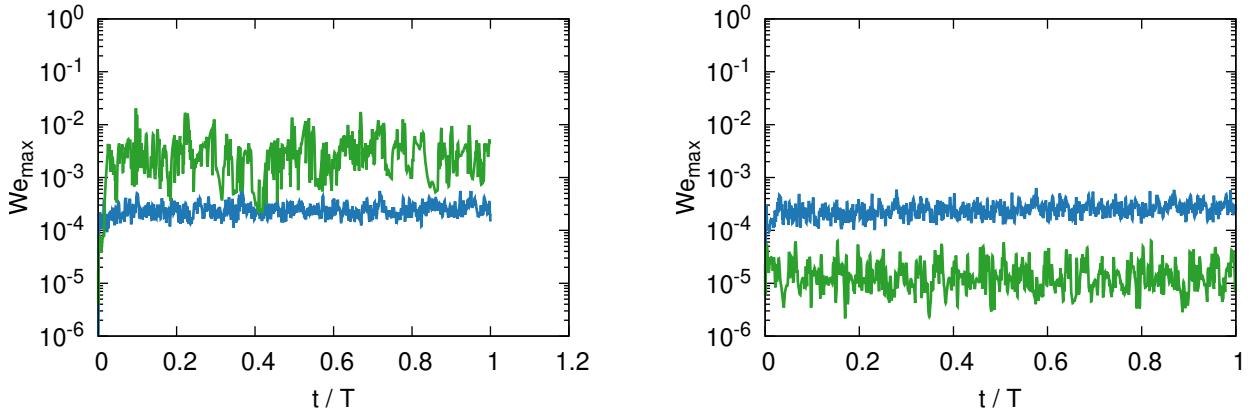


Figure 16: Evolution of the velocity error for a translating droplet in 3D at resolutions $R/h = 9.51$ (left) and 11.31 (right): present —— and Basilisk ——.

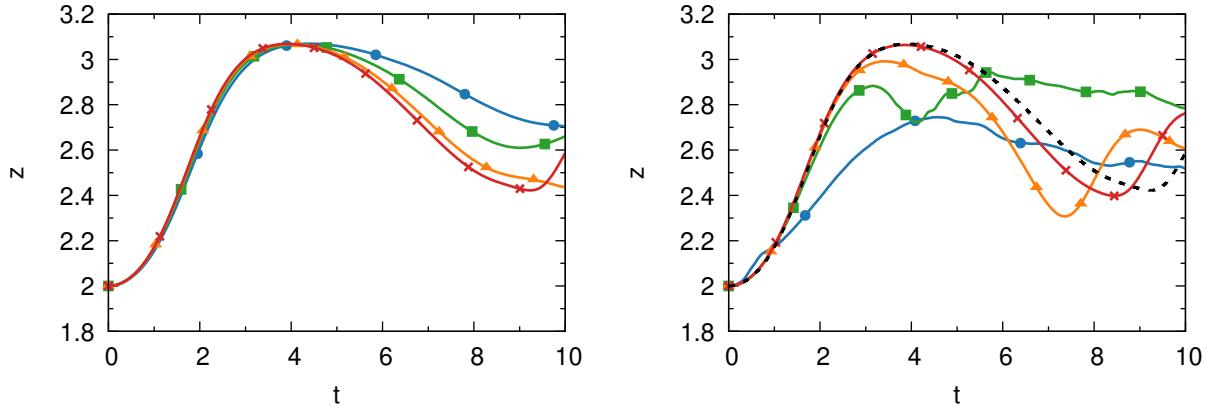


Figure 17: Trajectory of the bubble center of mass in the z -direction at various resolutions $R/h = 4.07$ (blue circles), 6.11 (green squares), 8.15 (orange triangles), and 12.23 (red crosses) – present (left) and Basilisk (right). The dashed line – - - on the right shows the trajectory at $R/h = 12.23$ from the present method.

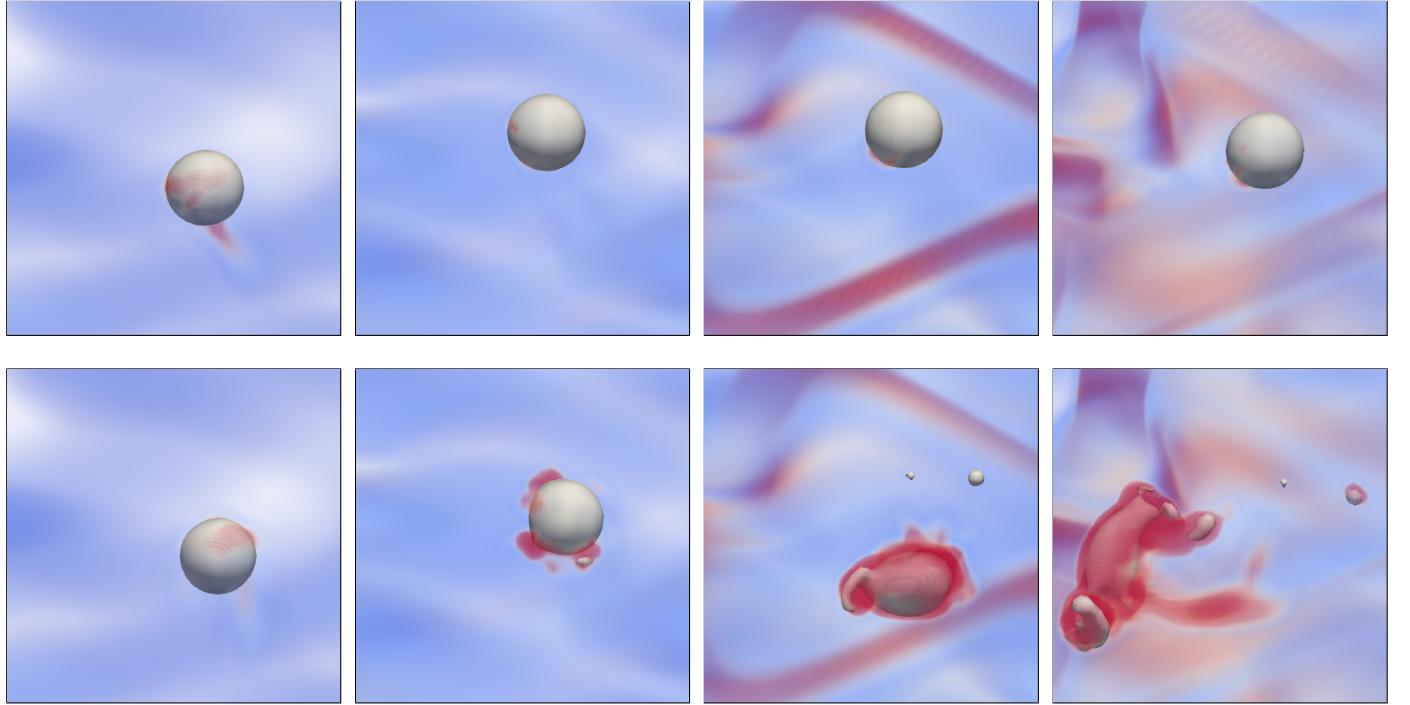


Figure 18: Isosurfaces of the volume fraction and the magnitude of vorticity (larger values with red) computed by the present method (top) and Basilisk (bottom) with $R/h = 6.11$ at $t = 2, 3, 4$ and 5 .

fraction on different resolutions at $t/T = 0.17$ and 0.34 , where $T = \sqrt{\rho_1 R^3 / \sigma}$ is the capillary time. The evolution of the neck radius is presented in Figure 21 in comparison to the experimental data reported in [31]. We found that our results match when a factor of 1.2 is applied to the values of time from the experiment. We observe that our method more accurately describes the shapes of bubbles compared to experimental data. Basilisk introduces spurious disturbances on the interface near the neck at $t/T = 0.17$ and the solution develops with a time lag.

5. Conclusion

We have presented a new method for estimating the curvature of the interface by fitting circular arcs to its piecewise linear reconstruction. The circular arcs are represented as strings of particles which evolve under constraints and forces attracting them to the interface.

The application of the method on a number of benchmark problems shows a significant improvement in the accuracy of computing the curvature of interfaces at low resolutions over the generalized height-function method [25] implemented in Basilisk [1]. The present method is more accurate at resolutions up to eight cells per radius and even with one cell per radius provides the relative curvature error below 0.1. We also demonstrate the capabilities of this hybrid method on a number of applications including multiphase vortical flows and bubble coalescence. Further applications include the bubble dynamics in electrochemical cells [15] and a plunging jet with air entrainment [17].

We note that at higher resolutions the present method shows only a first-order convergence for the absolute curvature, unlike the method of height functions which exhibits higher convergence rates. Ongoing work includes a switch to the use of the height functions if they can be evaluated in a sufficient number of points.

The present technique restricts the particles to circular arcs and computes the attraction force from the nearest point on the interface. However, the method allows for modifications

of the constraints and forces. For instance, the force can be computed directly from the volume fraction using the area cut by the string of particles. Forces defined from the intersection with the reconstructed volume can lead to an approach similar to the mesh-decoupled height functions [23]. Finally, the property of recovering the exact curvature mentioned in Section 2.2 can contribute to the existence of volume fraction fields providing a uniform curvature field and, therefore, exact equilibration of a static droplet. Such modifications constitute the subject of future work.

6. Acknowledgements

This research is funded by grant no. CRSII5_173860 of the Swiss National Science Foundation. The authors acknowledge the use of computing resources from CSCS (projects s754 and s931). We thank Professor Georges-Henri Cottet (Grenoble, France) for several helpful discussions regarding this work.

References

- [1] Basilisk. <http://basilisk.fr/>.
- [2] Cubism: Parallel block processing library. <https://gitlab.ethz.ch/mavt-cse/Cubism>.
- [3] HYPRE: Scalable linear solvers. <https://computation.llnl.gov/projects/hypre-scalable-linear-solvers-multigrid-methods>.
- [4] ANTHONY, C. R., KAMAT, P. M., THETE, S. S., MUNRO, J. P., LISTER, J. R., HARRIS, M. T., AND BASARAN, O. A. Scaling laws and dynamics of bubble coalescence. *Physical Review Fluids* 2, 8 (2017), 083601.
- [5] ASHBY, S. F., AND FALGOUT, R. D. A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations. *Nuclear Science and Engineering* 124, 1 (1996), 145–159.
- [6] AULISA, E., MANSERVISI, S., SCARDOVELLI, R., AND ZALESKI, S. Interface reconstruction with least-squares fit and split advection in three-dimensional cartesian geometry. *Journal of Computational Physics* 225, 2 (2007), 2301–2319.
- [7] BOISSONNEAU, P., AND BYRNE, P. An experimental investigation of bubble-induced free convection in a small electrochemical cell. *Journal of Applied Electrochemistry* 30, 7 (2000), 767–775.
- [8] BORNIA, G., CERVONE, A., MANSERVISI, S., SCARDOVELLI, R., AND ZALESKI, S. On the properties and limitations of the height function method in two-dimensional cartesian geometry. *Journal of Computational Physics* 230, 4 (2011), 851–862.

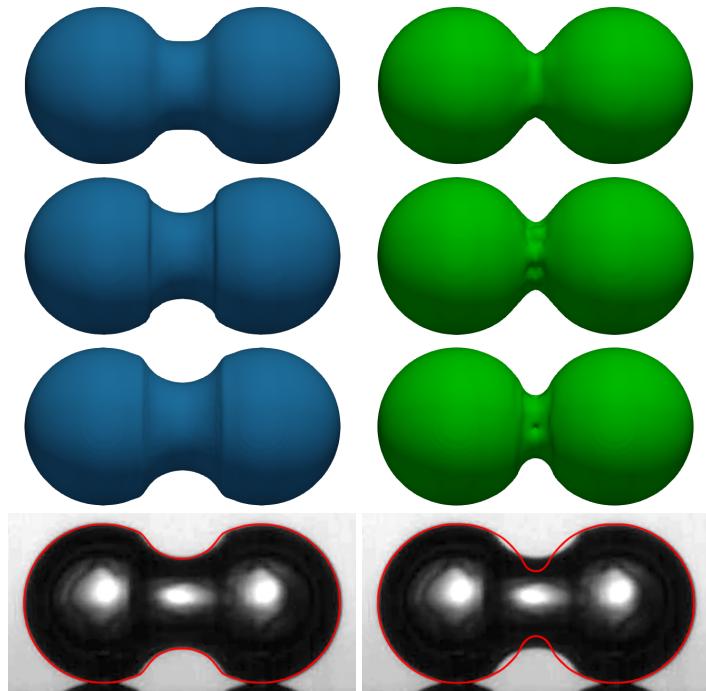


Figure 19: Isosurfaces of volume fraction at $t/T = 0.17$ depending on resolution $R/h = 9.6$ (top), 19.2 (middle) and 38.4 (bottom): present (left) and Basilisk (right). Experimental image [31] overlapped with contours from the finest mesh of each method.

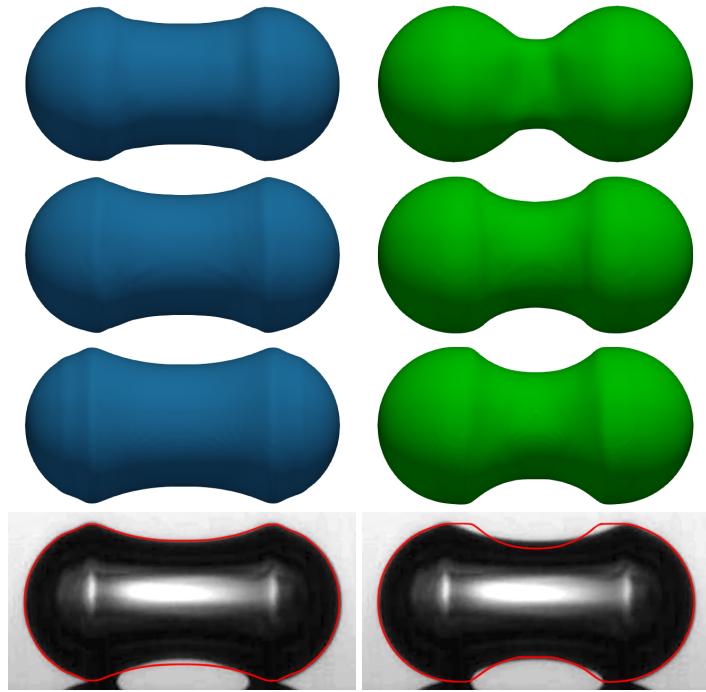


Figure 20: Isosurfaces of volume fraction at $t/T = 0.34$ depending on resolution $R/h = 9.6$ (top), 19.2 (middle) and 38.4 (bottom): present (left) and Basilisk (right). Experimental image [31] overlapped with contours from the finest mesh of each method.

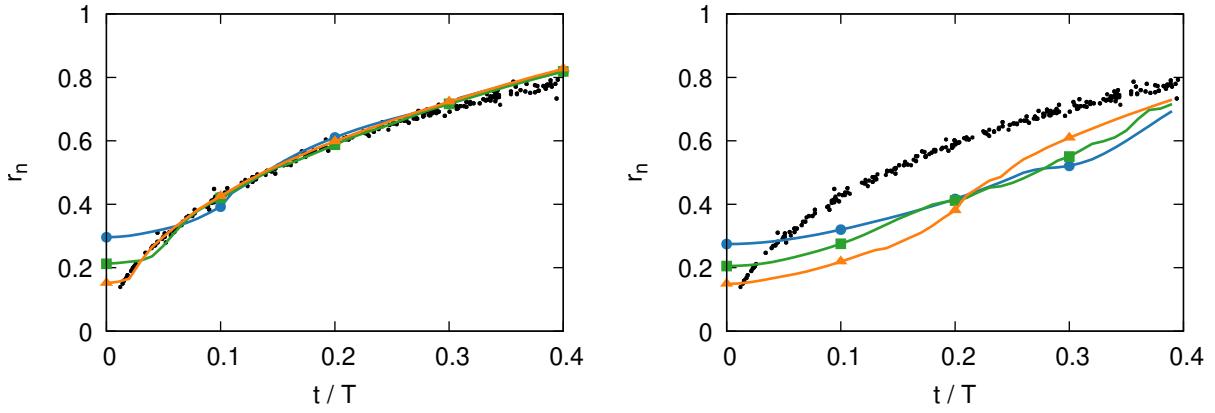


Figure 21: Evolution of the neck radius depending on the resolution $N_x = 9.6$ —●—, 19.2 —■—, 38.4 —▲—: present (left) and Basilisk (right) compared to experiment [31] (dots).

- [9] BRACKBILL, J. U., KOTHE, D. B., AND ZEMACH, C. A continuum method for modeling surface tension. *Journal of computational physics* 100, 2 (1992), 335–354.
- [10] CUMMINS, S. J., FRANCOIS, M. M., AND KOTHE, D. B. Estimating curvature from volume fractions. *Computers & structures* 83, 6-7 (2005), 425–434.
- [11] DIWAKAR, S., DAS, S. K., AND SUNDARARAJAN, T. A quadratic spline based interface (quasi) reconstruction algorithm for accurate tracking of two-phase flows. *Journal of Computational Physics* 228, 24 (2009), 9107–9130.
- [12] EVRARD, F., DENNER, F., AND VAN WACHEM, B. Estimation of curvature from volume fractions using parabolic reconstruction on two-dimensional unstructured meshes. *Journal of Computational Physics* 351 (2017), 271–294.
- [13] FALGOUT, R. D., AND YANG, U. M. hypre: A library of high performance preconditioners. In *International Conference on Computational Science* (2002), Springer, pp. 632–641.
- [14] FERZIGER, J. H., AND PERIC, M. *Computational methods for fluid dynamics*. Springer Science & Business Media, 2012.
- [15] HASHEMI, S. M. H., KARNAKOV, P., HADIKHANI, P., CHINELLO, E., LITVINOV, S., MOSER, C., KOUOMOUTSAKOS, P., AND PSALTIS, D. A versatile and membrane-less electrochemical reactor for the electrolysis of water and brine. *Energy & Environmental Science* (2019).
- [16] JAKOBSEN, H. A., LINDBORG, H., AND DORAO, C. A. Modeling of bubble column reactors: progress and limitations. *Industrial & engineering chemistry research* 44, 14 (2005), 5107–5151.
- [17] KARNAKOV, P., WERMELINGER, F., CHATZIMANOLAKIS, M., LITVINOV, S., AND KOUOMOUTSAKOS, P. A high performance computing framework for multiphase, turbulent flows on structured grids. In *Proceedings of the Platform for Advanced Scientific Computing Conference* (2019), ACM.
- [18] KASS, M., WITKIN, A., AND TERZOPOULOS, D. Snakes: Active contour models. *International journal of computer vision* 1, 4 (1988), 321–331.
- [19] KHARANGATE, C. R., AND MUDAWAR, I. Review of computational studies on boiling and condensation. *International Journal of Heat and Mass Transfer* 108 (2017), 1164–1196.
- [20] KIGER, K. T., AND DUNCAN, J. H. Air-entrainment mechanisms in plunging jets and breaking waves. *Annual Review of Fluid Mechanics* 44 (2012), 563–596.
- [21] LEONARD, B. P. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer methods in applied mechanics and engineering* 19, 1 (1979), 59–98.
- [22] LING, Y., ZALESKI, S., AND SCARDOVELLI, R. Multiscale simulation of atomization with small droplets represented by a lagrangian point-particle model. *International Journal of Multiphase Flow* 76 (2015), 122–143.
- [23] OWKES, M., AND DESJARDINS, O. A mesh-decoupled height function method for computing interface curvature. *Journal of Computational Physics* 281 (2015), 285–300.
- [24] PATANKAR, S. V., AND SPALDING, D. B. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. In *Numerical Prediction of Flow, Heat Transfer, Turbulence and Combustion*. Elsevier, 1983, pp. 54–73.
- [25] POPINET, S. An accurate adaptive solver for surface-tension-driven interfacial flows. *Journal of Computational Physics* 228, 16 (2009), 5838–5866.
- [26] POPINET, S. Numerical models of surface tension. *Annual Review of Fluid Mechanics* 50 (2018), 49–75.
- [27] RENARDY, Y., AND RENARDY, M. Prost: a parabolic reconstruction of surface tension for the volume-of-fluid method. *Journal of computational physics* 183, 2 (2002), 400–421.
- [28] SAAD, Y., AND SCHULTZ, M. H. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing* 7, 3 (1986), 856–869.
- [29] SCARDOVELLI, R., AND ZALESKI, S. Direct numerical simulation of free-surface and interfacial flow. *Annual review of fluid mechanics* 31, 1 (1999), 567–603.
- [30] SCARDOVELLI, R., AND ZALESKI, S. Analytical relations connecting linear

- interfaces and volume fractions in rectangular grids. *Journal of Computational Physics* 164, 1 (2000), 228–237.
- [31] SOTO, Á. M., MADDALENA, T., FRATERS, A., VAN DER MEER, D., AND LOHSE, D. Coalescence of diffusively growing gas bubbles. *Journal of Fluid Mechanics* 846 (2018), 143–165.
- [32] STROBL, S., FORMELLA, A., AND PÖSCHEL, T. Exact calculation of the overlap volume of spheres and mesh elements. *Journal of Computational Physics* 311 (2016), 158–172.
- [33] SUSSMAN, M., FATEMI, E., SMEREKA, P., AND OSHER, S. An improved level set method for incompressible two-phase flows. *Computers & Fluids* 27, 5-6 (1998), 663–680.
- [34] SUSSMAN, M., AND OHTA, M. High-order techniques for calculating surface tension forces. In *Free Boundary Problems*. Springer, 2006, pp. 425–434.
- [35] VAN REES, W. M., LEONARD, A., PULLIN, D., AND KOUMOUTSAKOS, P. A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high reynolds numbers. *Journal of Computational Physics* 230, 8 (2011), 2794–2805.
- [36] WERMELINGER, F., RASTHOFER, U., HADJIDOUKAS, P. E., AND KOUMOUTSAKOS, P. Petascale simulations of compressible flows with interfaces. *Journal of computational science* 26 (2018), 217–225.