A hybrid particle volume-of-fluid method for curvature estimation in multiphase flows

Petr Karnakov, Sergey Litvinov, Petros Koumoutsakos*

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

ARTICLE INFO

ABSTRACT

We present a particle method for estimating the curvature of interfaces in volume-of-fluid simulations of multiphase flows. The method is well suited for under-resolved interfaces, and it is shown to be more accurate than the parabolic fitting that is employed in such cases. The curvature is computed from the equilibrium positions of particles constrained to circular arcs and attracted to the interface. The proposed particle method is combined with the method of height functions at higher resolutions, and it is shown to outperform the current combinations of height functions and parabolic fitting. The algorithm is conceptually simple and straightforward to implement on new and existing software frameworks for multiphase flow simulations thus enhancing their capabilities in challenging flow problems. We evaluate the proposed hybrid method on a number of two- and three-dimensional benchmark flow problems and illustrate its capabilities on simulations of flows involving bubble coalescence and turbulent multiphase flows.

1. Introduction

Bubbles and drops are critical components of important industrial applications such as boiling and condensation (Kharangate and Mudawar, 2017), bubble column reactors (Jakobsen et al., 2005), electrochemical cells (Boissonneau and Byrne, 2000) and physical systems involving air entrainment in plunging jets (Kiger, Duncan, 2012) and liquid jet atomization (Ling et al., 2015). Simulations of such processes are challenged by the multiple scales of bubbles and their surface tension. Since the pioneering work of Brackbill et al. (1992) in modeling surface tension with the Eulerian representation, a number of advances have been made (Popinet, 2018), using level-sets (Sussman et al., 1998) and volume-of-fluid methodologies (VOF) (Scardovelli and Zaleski, 1999) 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 (Renardy and Renardy, 2002) and the balancing of pressure gradients with surface tension. The interface curvature estimation was further improved by the method of height functions (Cummins et al., 2005) 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 (Sussman and Ohta, 2006). 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 (Popinet, 2009). Modifications of the method aim to weaken this requirement by fitting an analytical function to the known values (Renardy and Renardy, 2002; Bornia et al., 2011; Diwakar et al., 2009). 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 (GHF) method (Popinet, 2009) which used parabolic fitting to heights from mixed directions and to centroids of the interface fragments. A similar approach was later implemented by Ling et al. (2015). An alternative approach is the mesh-decoupled height function method (Owkes and Desjardins, 2015) 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 reconstruc-

---

* Corresponding author.
E-mail address: petros@ethz.ch (P. Koumoutsakos).

tion directly from the volume fractions (Evrard et al., 2017) 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 (Kass et al., 1988). 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. Our approach is also different from the finite particle method (Wenzel and Garrick, 2018) which uses particles to construct a smooth representation of the volume fraction field: the particles are assigned with weights computed by averaging the volume fraction over neighboring cells, but their positions remain constant. On the other hand, out method only uses the positions of particles determined through an equilibration process. We note that the present algorithm is more accurate than the generalized height-function method (Popinet, 2009; Ling et al., 2015) up to a resolution of four cells per curvature radius in three dimensions and even at a resolution of one cell per radius provides the relative curvature error below 10%.

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

In this section, we describe a standard numerical model for two-component incompressible flows with surface tension and introduce our particle method for estimating the interface curvature.

2.1. VOF method for multiphase flows with surface tension

We consider the numerical model describing two-component incompressible flows with surface tension available in the open-source solver Basilisk (Popinet, 2009; Basilisk, 2014). The system consists of the Navier-Stokes equations for the mixture velocity $\boldsymbol{u}$ and pressure p

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

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

and the advection equation for the volume fraction α

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

with density $\rho = (1 - \alpha)\rho_1 + \alpha \rho_2$, dynamic viscosity $\mu = (1 - \alpha)\mu_1 + \alpha \mu_2$, gravitational acceleration $\boldsymbol{g}$ and constant material parameters ρ_1, ρ_2, μ_1 and μ_2. The surface tension force is defined as $\boldsymbol{f}_\sigma = \sigma \kappa \nabla \alpha$ with the surface tension coefficient σ and interface curvature κ. A finite volume discretization is based on Chorin's projection method for the pressure coupling (Chorin, 1968) and the Bell-Colella-Glaz scheme (Bell et al., 1989) for convective fluxes. The advection equation is solved using the volume-of-fluid method PLIC with piecewise linear reconstruction

(Weymouth and Yue, 2010), where the normals are computed using the mixed Youngs-centered scheme which is a combination of Youngs' scheme and the height functions. The approximation of the surface tension force is well-balanced (Francois et al., 2006) (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 computed as the average over the neighboring cells. We refer to Aniszewski et al. (2019) for further details about the algorithm.

To compute the cell-centered interface curvature, we use the proposed algorithm described in the following sections. Our implementation of the curvature estimator is available online[1] and can be directly used in Basilisk for simulations of multiphase flows with surface tension. We also provide a visual web-based demonstration of the method[2] and a reference implementation in Python.[3]

2.2. Particles for estimating the curvature from line segments

This section introduces a particle method for estimating the curvature of an interface represented by a set of line segments, which are given by the piecewise linear reconstruction of the interface as described in the next section. The particles are constrained to circular arcs and equilibrate on the interface due to attraction forces.

Consider a set of N_L line segments $[\boldsymbol{a}_l, \boldsymbol{b}_l]$, $l = 1, \ldots, N_L$ with endpoints $\boldsymbol{a}_l, \boldsymbol{b}_l \in \mathbb{R}^2$. One is distinguished as the target line segment $[\boldsymbol{a}^*, \boldsymbol{b}^*]$ at which the curvature is to be estimated. The union of all line segments is denoted as

$$L = \bigcup_{l=1}^{N_L} [\boldsymbol{a}_l, \boldsymbol{b}_l]. \tag{4}$$

Given an odd number N, we introduce a string of N particles $\boldsymbol{x}_i \in \mathbb{R}^2$, $i = 1, \ldots, N$ and denote the index of the central particle as $c = (N + 1)/2$. The particles are constrained to circular arcs, which leads to parametrization of their positions

$$\boldsymbol{x}_i(\boldsymbol{p}, \phi, \theta) = \begin{cases} \boldsymbol{p} + \sum_{j=1}^{i-c} h_p \boldsymbol{e}(\phi + (j - \frac{1}{2})\theta) & i > c, \\ \boldsymbol{p} & i = c, \\ \boldsymbol{p} - \sum_{j=1}^{c-i} h_p \boldsymbol{e}(\phi - (j - \frac{1}{2})\theta) & i < c, \end{cases} \tag{5}$$

where $\boldsymbol{p}$ is the origin, ϕ is the rotation angle, θ is the bending angle, $h_p = H_p h / (N - 1)$ is the distance between neighboring particles and $\boldsymbol{e}(\psi) = \cos \psi \, \boldsymbol{e}_x + \sin \psi \, \boldsymbol{e}_y$. Parameter H_p defines the length of the string relative to the mesh step h. Values of N and H_p are chosen in Section 2.7. The curvature of the circular arc is related to the bending angle as

$$\kappa(\theta) = \frac{2 \sin \frac{\theta}{2}}{h_p}. \tag{6}$$

We define the force attracting a particle at position $\boldsymbol{x}$ to the nearest point on the interface

$$\boldsymbol{f}(\boldsymbol{x}) = \eta \, (\underset{\boldsymbol{y} \in L}{\arg \min} |\boldsymbol{y} - \boldsymbol{x}| - \boldsymbol{x}), \tag{7}$$

where $\eta \in [0, 1]$ is a relaxation factor. Fig. 1 illustrates the parametrisation of positions and computation of forces.

Further derivations use vector notation of the form $\boldsymbol{X} = [\boldsymbol{x}_i, i = 1, \ldots, N]$, where $\boldsymbol{x}_i$ is component i of $\boldsymbol{X}$. The scalar product is defined as $\boldsymbol{X} \cdot \boldsymbol{Y} = \sum_{i=1}^{N} \boldsymbol{x}_i \cdot \boldsymbol{y}_i$. In this notation, the positions and forces
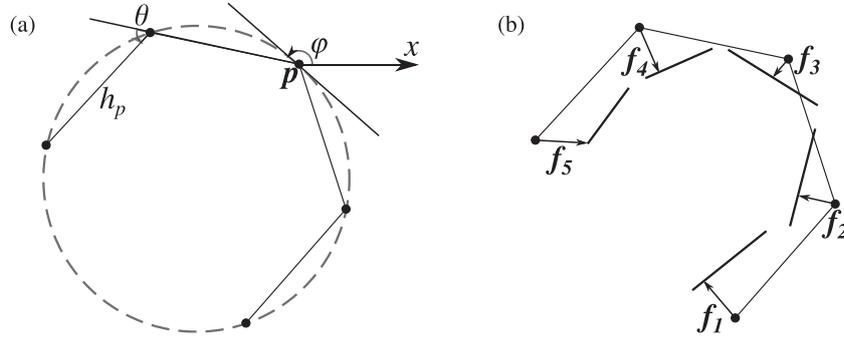
Fig. 1. (a) All particles belong to a circle, and their positions are defined by the central particle $\boldsymbol{p}$, the orientation angle ϕ and the bending angle θ. (b) Line segments represent the interface, and the force acts on each particle towards the nearest point on the interface.

combine to

$$\boldsymbol{X}(\boldsymbol{p},\phi,\theta) = [\boldsymbol{x}_i(\boldsymbol{p},\phi,\theta),\ i=1,\dots,N], \tag{8}$$

$$\boldsymbol{F}(\boldsymbol{p},\phi,\theta) = [\boldsymbol{f}(\boldsymbol{x}_i(\boldsymbol{p},\phi,\theta)),\ i=1,\dots,N]. \tag{9}$$

The particles evolve until equilibration according to the following iterative procedure. Initially, the particles are arranged along the target line segment $[\boldsymbol{a}^*,\boldsymbol{b}^*]$. The central particle is placed at the segment center $\boldsymbol{p}^0 = (\boldsymbol{a}^* + \boldsymbol{b}^*)/2$, the rotation angle ϕ^0 is given by vector $\boldsymbol{b}^* - \boldsymbol{a}^*$, and the bending angle is zero $\theta^0 = 0$. Iteration $m+1$ consists of three steps, each correcting one parameter:

Step 1.

Compute positions and forces, correct $\boldsymbol{p}$ by the force on the central particle and subtract the change of positions from the forces

$$\boldsymbol{X}^m = \boldsymbol{X}(\boldsymbol{p}^m,\phi^m,\theta^m), \quad \boldsymbol{F}^m = \boldsymbol{F}(\boldsymbol{p}^m,\phi^m,\theta^m)$$

$$\boldsymbol{p}^{m+1} = \boldsymbol{p}^m + \boldsymbol{f}_c^m$$

$$\boldsymbol{X}^* = \boldsymbol{X}(\boldsymbol{p}^{m+1},\phi^m,\theta^m), \quad \boldsymbol{F}^* = \boldsymbol{F}^m - (\boldsymbol{X}^* - \boldsymbol{X}^m).$$

Step 2.

Correct ϕ by projection of force on derivative and subtract the change of positions from the forces

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

$$\phi^{m+1} = \phi^m + \frac{\boldsymbol{F}^* \cdot \boldsymbol{D}_\phi}{\boldsymbol{D}_\phi \cdot \boldsymbol{D}_\phi}$$

$$\boldsymbol{X}^{**} = \boldsymbol{X}(\boldsymbol{p}^{m+1},\phi^{m+1},\theta^m), \quad \boldsymbol{F}^{**} = \boldsymbol{F}^* - (\boldsymbol{X}^{**} - \boldsymbol{X}^*).$$

Step 3.

Correct θ by projection of force on derivative

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

$$\theta^{m+1} = \theta^m + \frac{\boldsymbol{F}^{**} \cdot \boldsymbol{D}_\theta}{\boldsymbol{D}_\theta \cdot \boldsymbol{D}_\theta}.$$

Expressions for derivatives $\frac{\partial \boldsymbol{X}}{\partial \phi}$ and $\frac{\partial \boldsymbol{X}}{\partial \theta}$ are provided in Section A.3, and a proof of convergence is given in Section A.4.

The iterations are repeated until

$$E_m = \frac{\max_i |\boldsymbol{x}_i^m - \boldsymbol{x}_i^{m-1}|_\infty}{\eta\,h} < \varepsilon_p \quad \text{or} \quad m > m_{\max}, \tag{10}$$

where E_m is the maximum difference after iteration m, h is the mesh step, η is the relaxation parameter from (7), ε_p is the convergence tolerance and $m_{\max}$ is the maximum number of iterations. Values for these parameters are chosen in Section 2.4. Finally, the curvature is computed from the bending angle θ using relation (6).

Steps 2 and 3 are defined to provide the correction of positions which minimizes the distance to forces. With linear approximation in terms of $\phi^{m+1} - \phi^m$, this leads to a minimization problem

$$\left\| \boldsymbol{F}^* - (\phi^{m+1} - \phi^m)\boldsymbol{D}_\phi \right\|_2 \to \min \tag{11}$$

and gives the optimal correction

$$\phi^{m+1} = \phi^m + \frac{\boldsymbol{F}^* \cdot \boldsymbol{D}_\phi}{\boldsymbol{D}_\phi \cdot \boldsymbol{D}_\phi}. \tag{12}$$

The same procedure is applied to the bending angle θ. We note that using the same principle for Step 1 would correspond to correcting the origin $\boldsymbol{p}$ by the mean force $\frac{1}{N}\sum_{i=1}^N \boldsymbol{f}_i^m$ instead of $\boldsymbol{f}_c^m$. However, simulations on the test case with a static droplet introduced in Section 3.2 showed that this would result in stronger spurious flow and lack of equilibrium.

We also observe that, for the chosen initial conditions and forces, Step 1 trivializes to $\boldsymbol{p}^m = \boldsymbol{p}^0$ since the origin already belongs to a line segment, which results in zero force $\boldsymbol{f}_c$. However, we still include this step in the algorithm to allow for modifications of the attraction force. One such modification, described in Section A.1, consists in replacing the line segments with circular arcs and recovers the exact curvature if the endpoints of all line segments belong to a circle.

2.3. Particles for estimating the curvature from volume fraction field

We estimate the interface curvature from a discrete volume fraction field defined on a uniform Cartesian mesh. Our method uses a piecewise linear reconstruction of the interface, which is available as part of the PLIC technique for solving the advection equation. Following Aulisa et al. (2007), we compute the interface normals with the mixed Youngs-centered method on a $3 \times 3 \times 3$ stencil in 3D (or 3×3 in 2D). To define the orientation, we assume that the normals have the direction of anti-gradient of the volume fraction. Then, the interface is reconstructed in each cell independently by a polygon (or line segment) cutting the cell into two parts according to the estimated normal and the given volume fraction (Scardovelli and Zaleski, 2000).

In two dimensions, the interface is represented as a set of line segments. To estimate the curvature in one interfacial cell, we collect the line segments from a 5×5 stencil centered at the target cell. The stencil of this size includes all line segments that can be reached by the particles if the string length is $H_p \leq 4$. We apply the algorithm from Section 2.2 to these line segments and compute the curvature.

In three dimensions, the reconstructed interface is a set of planar convex polygons. In this case, we compute the mean curvature as the average over N_s 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:
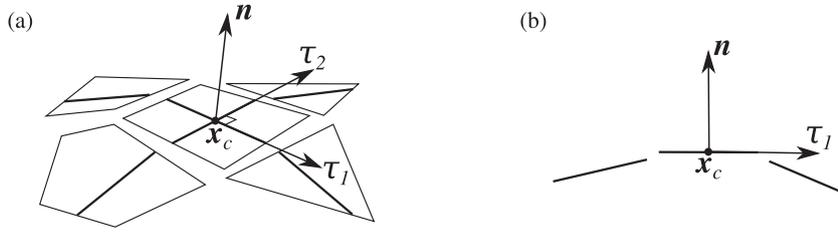
Fig. 2. Curvature estimation in three dimensions in case $N_s = 2$. (a) Interface polygons with the cross sections. (b) One cross section in local two-dimensional coordinates.

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 $\boldsymbol{n}$ of the target polygon and the center $\boldsymbol{x}_c$ as the mean over its vertices.

Step 2.

Compute the curvature κ_j in each cross section $j = 0, \ldots, N_s - 1$:

- Define a plane passing through $\boldsymbol{x}_c$ and containing vectors $\boldsymbol{n}$ and
  $$\boldsymbol{\tau} = \cos(\pi j / N_s)\boldsymbol{\tau}_1 + \sin(\pi j / N_s)\boldsymbol{\tau}_2,$$
  where $\boldsymbol{\tau}_1 = \boldsymbol{n} \times \boldsymbol{e}$, $\boldsymbol{\tau}_2 = \boldsymbol{n} \times \boldsymbol{\tau}_1$ and $\boldsymbol{e}$ is one of the unit vectors $\{\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z\}$ providing the minimal $|\boldsymbol{n} \cdot \boldsymbol{e}|$.
- Intersect each polygon in P with the plane and collect non-empty intersections in a set of line segments $\{[\boldsymbol{a}_l, \boldsymbol{b}_l]\}$.
- Construct a set $\{[\widehat{\boldsymbol{a}}_l, \widehat{\boldsymbol{b}}_l]\}$ from $\{[\boldsymbol{a}_l, \boldsymbol{b}_l]\}$ by computing local two-dimensional coordinates of the endpoints
  $$\widehat{\boldsymbol{x}} = \big((\boldsymbol{x} - \boldsymbol{x}_c) \cdot \boldsymbol{\tau}, (\boldsymbol{x} - \boldsymbol{x}_c) \cdot \boldsymbol{n}\big) \in \mathbb{R}^2,$$
  where $\boldsymbol{x} = \boldsymbol{a}_l$ or $\boldsymbol{x} = \boldsymbol{b}_l$.
- Apply the procedure from Section 2.2 to $\{[\widehat{\boldsymbol{a}}_l, \widehat{\boldsymbol{b}}_l]\}$ and compute the curvature κ_j.

Step 3.

Compute the mean curvature

$$\kappa = \frac{1}{N_s} \sum_{j=0}^{N_s - 1} \kappa_j. \tag{13}$$

Fig. 2 illustrates the algorithm. This approach of cross-sections allows us to estimate the mean curvature in three dimensions by using the particle method formulated for line segments on a plane.

2.4. Convergence of iterations

The iterative algorithm in Section 2.2 is guaranteed to converge for a sufficiently small relaxation factor as shown in Section A.4. Here we demonstrate the convergence on the test case of estimating the curvature of a sphere introduced in Section 3.1. The convergence tolerance is set to $\varepsilon_p = 10^{-5}$ which is three orders of magnitude smaller than the typical curvature error. Fig. 3 shows the number of iterations required to satisfy the convergence criterion (10) depending on the relaxation factor. We choose the relaxation factor of $\eta = 0.5$ for which the equilibration takes 10–20 iterations as seen from the convergence history in Fig. 3. All further computations are performed with $\eta = 0.5$, $\varepsilon_p = 10^{-5}$ and $m_{\max} = 20$.

2.5. Generalized height-function method

We compare our method with the generalized height-function method for curvature estimation implemented in Basilisk[4]

---

[4] We use the version of Basilisk available as of August 21, 2019: http://basilisk.fr
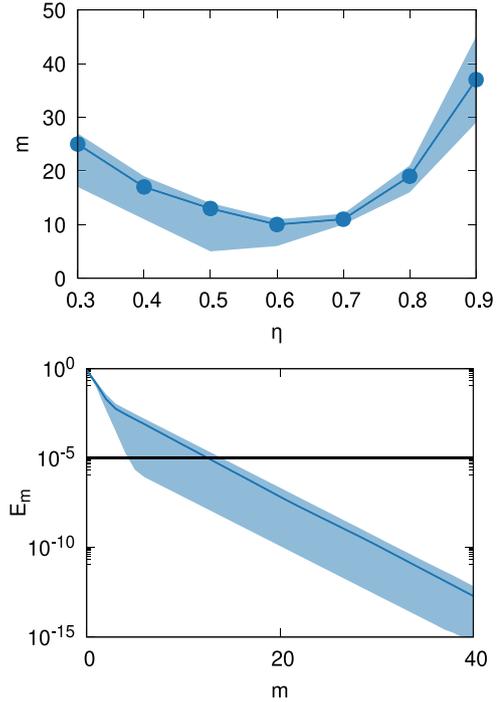
Fig. 3. Equilibration of particles on a sphere at resolution $R/h = 4$. The number of iterations required to reach the convergence criterion (10) with $\varepsilon_p = 10^{-5}$ depending on the relaxation factor η (top) and the convergence history for the difference after one iteration with $\eta = 0.5$ (bottom). The lines show the median and the shades show the 10% and 90% percentiles over the interfacial cells.

(Popinet, 2009; Basilisk, 2014). The algorithm attempts to compute the curvature in each interfacial cell with a series of techniques depending on the resolution. In three dimensions they can be outlined as follows

- Evaluate the height function along a preferred coordinate plane chosen based on the interface normal. The heights are obtained by summation over columns that span up to 13 cells. If consistent heights are available on the 3×3 stencil, compute the curvature from them with finite differences.
- Collect heights from mixed directions in the $3 \times 3 \times 3$ stencil. If found six consistent heights, compute the curvature from a paraboloid fitted to them.
- If the curvature is already defined in one of the neighboring cells in the $3 \times 3 \times 3$ stencil, copy the curvature from the neighbor.
- Collect centroids of interfacial cells in the $3 \times 3 \times 3$ stencil. If found six centroids, compute the curvature from a paraboloid fitted to them.
- If all techniques fail, set the curvature to zero.

Fig. 4. Curvature error for a 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 ━✕━.

We refer to the documentation of Basilisk[5] for further details of the algorithm. As clearly seen from the above description, the generalized height-function method, while providing an algorithm applicable at all resolutions, involves four different sources of curvature: heights, parabolic fitting to heights from mixed directions, values of curvature from neighboring cells and parabolic fit to centroids of the piecewise linear interface. Such procedure is complex for implementation and lacks robustness (e.g. depends on the presence of neighboring interfacial cells for parabolic fitting).

2.6. Combined particle and height-function method

As shown in Section 3, the error of our method asymptotically approaches a constant. To achieve second-order convergence, we follow the idea of the generalized height-function method and switch to standard heights at high resolutions. Our combined method therefore consists of two steps

- Evaluate the height function along a preferred coordinate plane chosen based on the interface normal. If consistent heights are available on a 3×3 stencil, compute the curvature from them with finite differences.
- Otherwise, compute the curvature using the proposed particle method as described in Section 2.3.

Overall, the logic of the algorithm has greatly simplified compared to the generalized height function method. At the same time, as we demonstrate in the following sections, the combined method provides better accuracy at low resolutions and second-order convergence for well-resolved interfaces.

2.7. 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 string length H_p. We set the parameters to $N = 7$, $N_s = 3$ and $H_p = 4$. Then, we vary each parameter independently and examine their influence by estimating the curvature of a sphere at various resolutions following the test case in Section 3.1. The figures present the median error over 100 samples for the center from a uniform distribution over the octant of the cell, i.e. each coordinate is sampled from the range $[0, h/2]$. As seen from Fig. 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 Fig. 5 is also small which is expected for a sphere. However, more complex shapes such as those observed during the bubble coalescence in Section 4.3, require at least $N_s = 3$ as shown in Fig. 7. Fig. 6 shows a stronger

Table 1
Runtime of the Taylor-Green vortex with 890 bubbles at resolution $R/h = 4$ on a mesh of 256^3. The benchmark is executed on Piz Daint with 512 cores. The runtime of one time step is averaged over all steps, and the number of interfacial cells is measured at the final time $t = 10$.

	present	GHF
Total time, per step	403 ms	329 ms
Curvature estimation time, per step	125 ms	33 ms
Curvature estimation time, to total time	31.1%	10.2%
Cells containing interface, to all cells	5.17%	5.04%
Cells with curvature from height functions, to all cells	0.69%	0.65%

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.

2.8. Computational cost

We compare the computational cost of our method to that of the generalized height-function method on the test case with a Taylor-Green vortex introduced in Section 4.1. The formulation is modified to consider 890 spherical bubbles at resolution $R/h = 4$ on a mesh of 256^3 cells. The initial positions of bubbles are distributed uniformly in the domain. The computations have been performed in parallel with 512 cores on Piz Daint supercomputer, where each compute node is equipped with a 12-core CPU Intel-®Xeon-®E5-2690 v3. Table 1 reports the runtime for both methods. About 5% of the cells contain interface fragments and require the curvature estimation. The time spent on the curvature estimation with the present method amounts to 125 ms per step or 31.1% of total time, while GHF takes 33 ms per step or 10.2% of total time. Therefore, our method is about four times more computationally expensive than GHF.

The cost of our method depends on the number of iterations and the number of particles per cell, which is controlled by parameters N, N_s and ε_p. Estimation of curvature for a given volume fraction field consists of two parts: extraction of line segments and iterations. Extraction of the line segments from neighboring cells takes about 40% of the time and scales linearly with N_s. Iterations for equilibration of particles take about 60% of the time and scale linearly with $N_s N$.

3. Test cases

We examine the capabilities of the proposed method 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) (Strobl et al., 2016).

---

[5] Curvature estimation in Basilisk: http://basilisk.fr/src/curvature.h
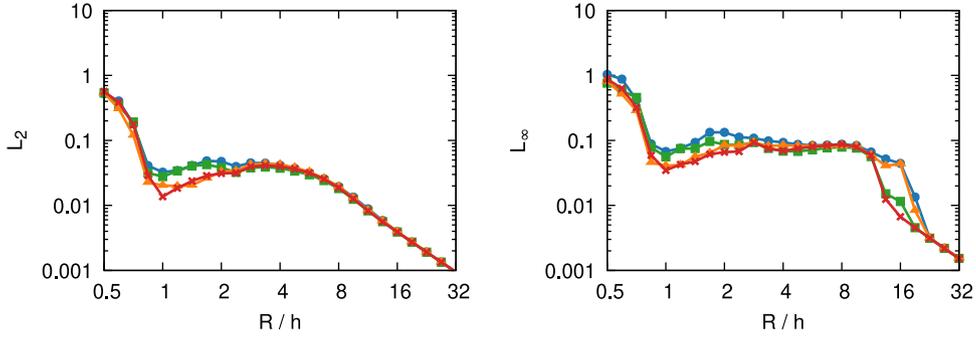
Fig. 5. Curvature error for a sphere in L_2 and L_∞ norms depending on the resolution for various values of the number of cross sections $N_s = 1$ ──●──, 2 ──■──, 3 ──▲── and 4 ──✕──.
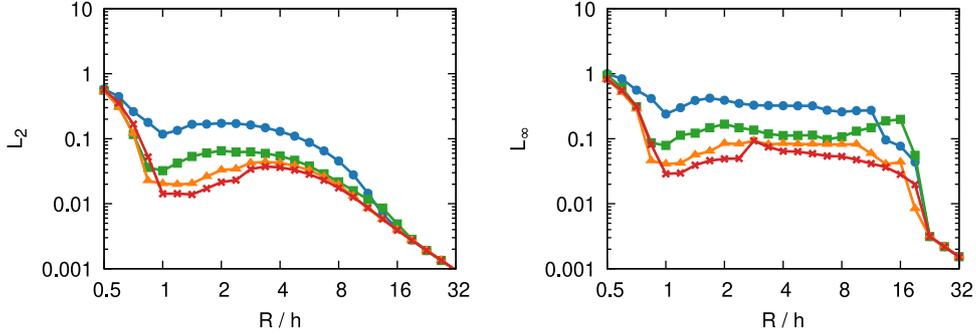
Fig. 6. Curvature error for a sphere in L_2 and L_∞ norms depending on the resolution for various values of the particle string length $H_p = 2$ ──●──, 3 ──■──, 4 ──▲── and 5 ──✕──.
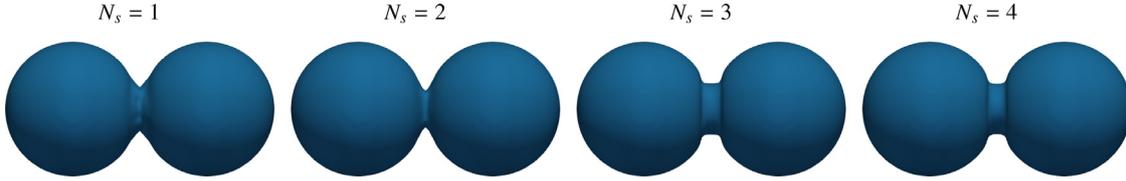
Fig. 7. Isosurfaces of the volume fraction with $R/h = 19.2$ at $t/T = 0.08$ depending on the number of cross sections $N_s = 1$, 2, 3 and 4.

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

$$L_2(\kappa) = \left(\frac{1}{|I|} \sum_{i \in I} \left(\frac{\kappa_i - \kappa_{\mathrm{ex}}}{\kappa_{\mathrm{ex}}} \right)^2 \right)^{1/2}, \tag{14}$$

$$L_\infty(\kappa) = \max_{i \in I} \left| \frac{\kappa_i - \kappa_{\mathrm{ex}}}{\kappa_{\mathrm{ex}}} \right|, \tag{15}$$

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. $\kappa_{\mathrm{ex}} = 2/R$ for sphere and $\kappa_{\mathrm{ex}} = 1/R$ for circle).

Fig. 10 shows the error in comparison to GHF. 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. The method of particles alone as described in Section 2.3 does not converge at high resolutions, and the maximum error saturates at about 10%. The combined method from Section 2.6 switches to height functions at high resolutions and therefore converges with second order. Figs. 8 and 9 show examples of final configuration of particles at resolutions below two cells per radius.

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, inertial and viscous forces

$$\mathrm{La} = \frac{2\sigma\rho R}{\mu^2}, \tag{16}$$

which we set to $\mathrm{La} = 1200$. 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 $p_L = \sigma\kappa_{\mathrm{ex}}$. The numerical solutions develop spurious currents due to an imbalance between the pressure gradient and surface tension. We compute the magnitude of the spurious velocity in the x-direction

$$U_{\max} = \max_{i \in C} |u_{x,i}|, \tag{17}$$
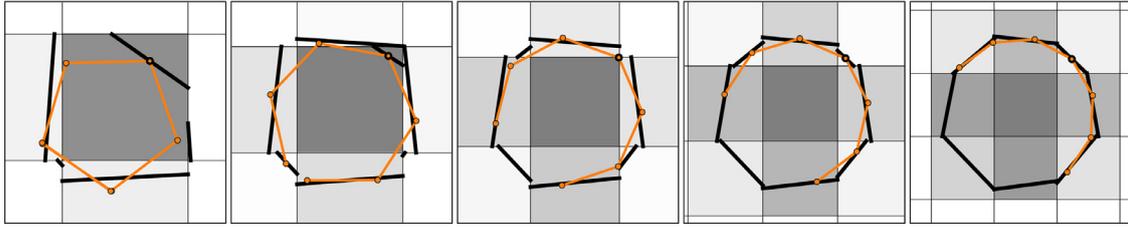
Fig. 8. Line segments of the interface (black) and positions of particles (orange) from one selected cell for a circle at resolutions $R/h = 0.59$, 0.71, 0.84, 1.0 and 1.19. The central particle is highlighted by a thicker edge. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
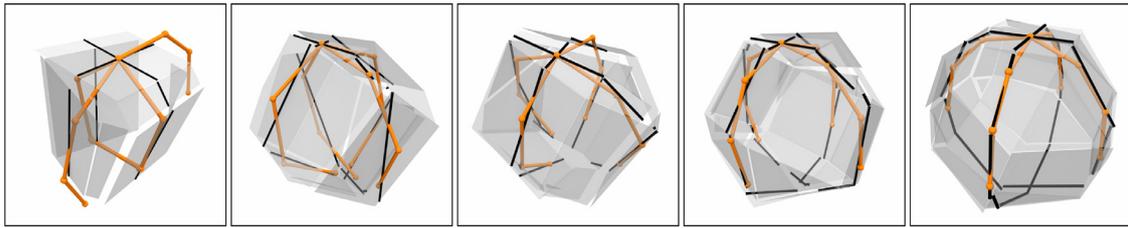
Fig. 9. Polygons of the interface, cross sections (black) in case $N_s = 2$ and positions of particles (orange) from one selected cell for a sphere at resolutions $R/h = 0.59$, 0.71, 0.84, 1.0 and 1.19. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
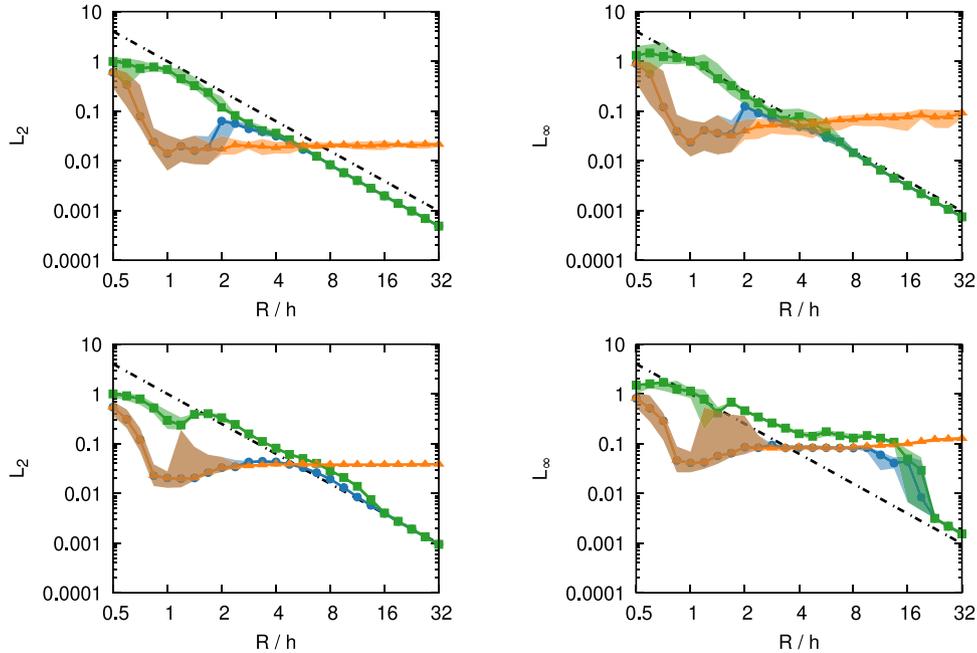
Fig. 10. Curvature error for a circle (top) and sphere (bottom) in L_2 and L_∞ norms depending on the resolution: particles only ——▲——, combined particles and heights ——●——, GHF ——■—— and second-order convergence ·—··—··—·· The lines show the median and the shades show the 10% and 90% percentiles for random positions of the center.

the corresponding Weber number

$$We_{max} = \frac{2\rho R U_{max}^2}{\sigma} \tag{18}$$

and the pressure jump

$$\Delta p = \max_{i \in C} p_i - \min_{i \in C} p_i, \tag{19}$$

where C is the indices of all cells.

Figs. 11 and 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 Fig. 13. We observe that the solutions from GHF convergence with time to a zero spurious flow in most cases. This demonstrates the existence of a volume fraction field for which the method of

height functions provides a uniform curvature field. In other cases, such as $R/h = 2$ and 26.9 in 3D, the spurious flow does not converge to zero. Meanwhile, our method of particles more accurately predicts the pressure jump at low resolutions below $R/h = 4$ cells in 3D (and 3.36 cells in 2D) and provides lower magnitude of the spurious flow in cases such as $R/h = 2$ in 3D shown in Fig. 13.

We note, however, that the equilibration of a static droplet is largely of theoretical interest as the equilibrium shape is symmetric and aligned with the mesh directions. Such conditions are incompatible with advection and therefore rarely observed in practically relevant simulations. The spurious flow occurs in a more realistic scenario such as the translating droplet case discussed in Section 3.3.
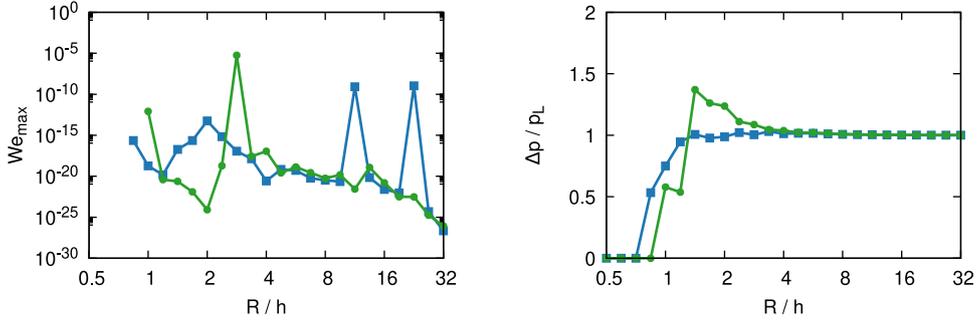
Fig. 11. Velocity error and the pressure jump for a static droplet in 2D depending on the resolution: present ▬■▬ and GHF ●.
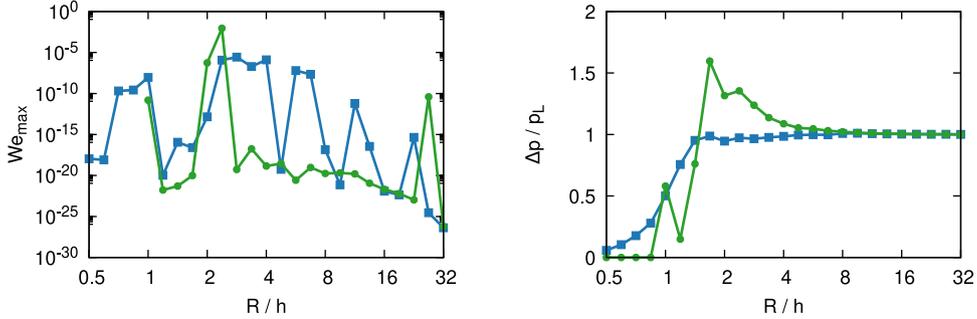
Fig. 12. Velocity error and the pressure jump for a static droplet in 3D depending on the resolution: present ▬■▬ and GHF ●.
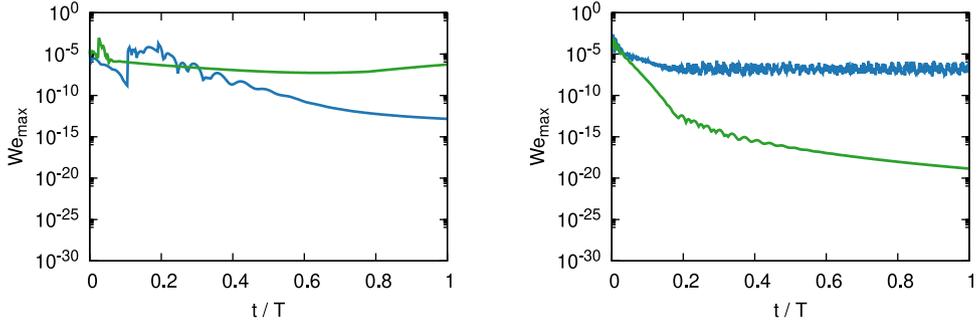
Fig. 13. Evolution of the velocity error for a static droplet in 3D at resolutions $R/h = 2$ (left) and 4 (right): present ———— and GHF ————.

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 and $\mathbf{l} = (1, 0.8)$ in 2D. The additional parameter of the problem is the Weber number

$$\text{We} = \frac{2\rho R U^2}{\sigma}, \tag{20}$$

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

$$U_{\max} = \max_{i \in C} |u_{x,i} - U_x|, \tag{21}$$

and definitions of $\text{We}_{\max}$ and Δp follow (18) and (19). Figs. 14 and 15 show $\text{We}_{\max}$ 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 $\text{We}_{\max}$ for two selected resolutions is shown in Fig. 16. Our method provides lower magnitudes of the spurious flow than GHF and more accurate values of the pressure jump at resolutions below $R/h = 2.82$ cells in 3D (and $R/h = 2$ cells in 2D),

and at higher resolutions the error is comparable to GHF. We note that more accurate estimates of curvature in Section 3.1 are observed at similar resolutions.

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 (Van Rees et al., 2011). Here we extend the formulation by adding a gaseous phase. The problem is solved in a periodic domain [0, 2π]^3 with the initial velocity

$$\begin{aligned}
u_x &= \sin x \, \cos y \, \cos z, \\
u_y &= -\cos x \, \sin y \, \cos z, \\
u_z &= 0
\end{aligned} \tag{22}$$

and a single bubble of radius $R = 0.1$ 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 $\text{We} = 2$. The density and viscosity ratios are set to $\rho_2/\rho_1 = 0.01$ and $\mu_2/\mu_1 = 0.01$.

Fig. 17 shows the trajectory of the bubble at various resolutions in comparison to GHF. Both methods converge to the
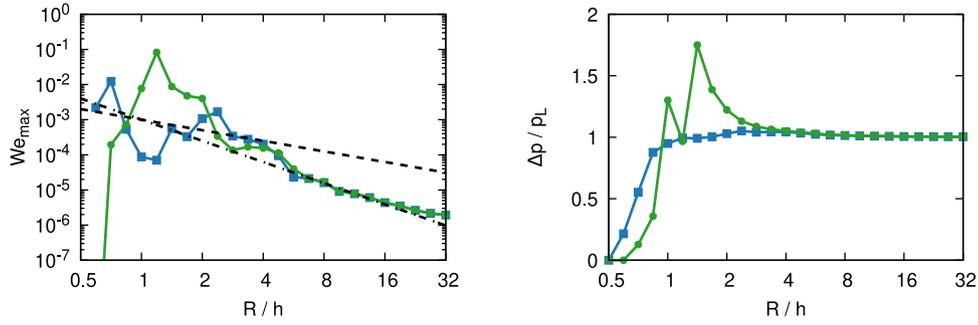
Fig. 14. Velocity error and the pressure jump for a translating droplet in 2D depending on the resolution: present ─■─ and GHF ─●─. Convergence with first ----- and second ·-··-··-·· order.
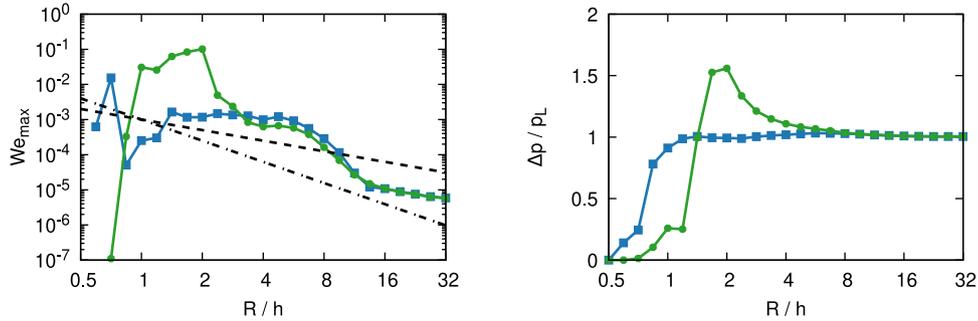
Fig. 15. Velocity error and the pressure jump for a translating droplet in 3D depending on the resolution: present ─■─ and GHF ─●─. Convergence with first ----- and second ·-··-··-·· order.
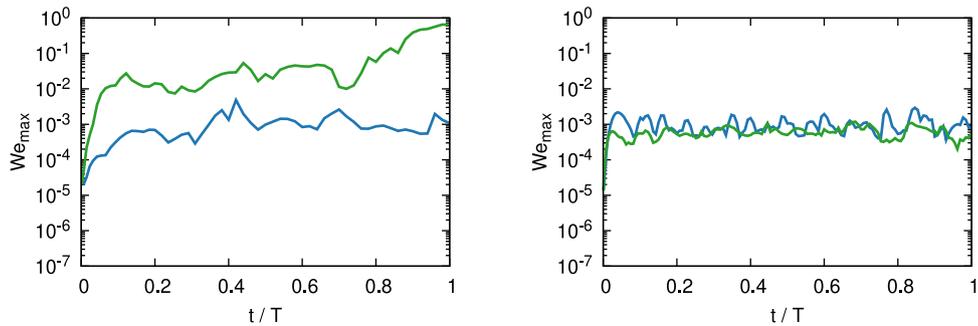
Fig. 16. Evolution of the velocity error for a translating droplet in 3D at resolutions $R/h = 2$ (left) and 4 (right): present ──── and GHF ────.
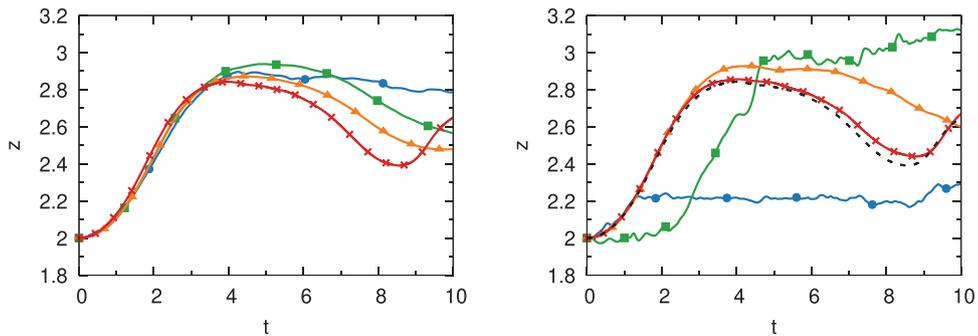
Fig. 17. Trajectory of the bubble center of mass in the z-direction at various resolutions $R/h = 1.53$ ─●─, 2.04 ─■─, 3.06 ─▲─ and 4.08 ─✕─ : present (left) and GHF (right). The dashed line ----- on the right shows the trajectory at $R/h = 4.08$ from the present method.

same solution with the mesh refinement. However, at lower resolutions $R/h \leq 3.06$ our method provides qualitatively more accurate trajectories, while in GHF the trajectory is dominated by the spurious flow. Snapshots of the vorticity magnitude and the bubble shape are shown for both methods at $R/h = 3.06$ in Fig. 18.

4.2. Taylor-Green vortex with elongated droplet

To evaluate our method on non-spherical shapes, we consider the Taylor-Green vortex with a droplet initially elongated in the z-direction. We use the same initial conditions (22) for the velocity field and define the volume fraction to describe a single elliptical
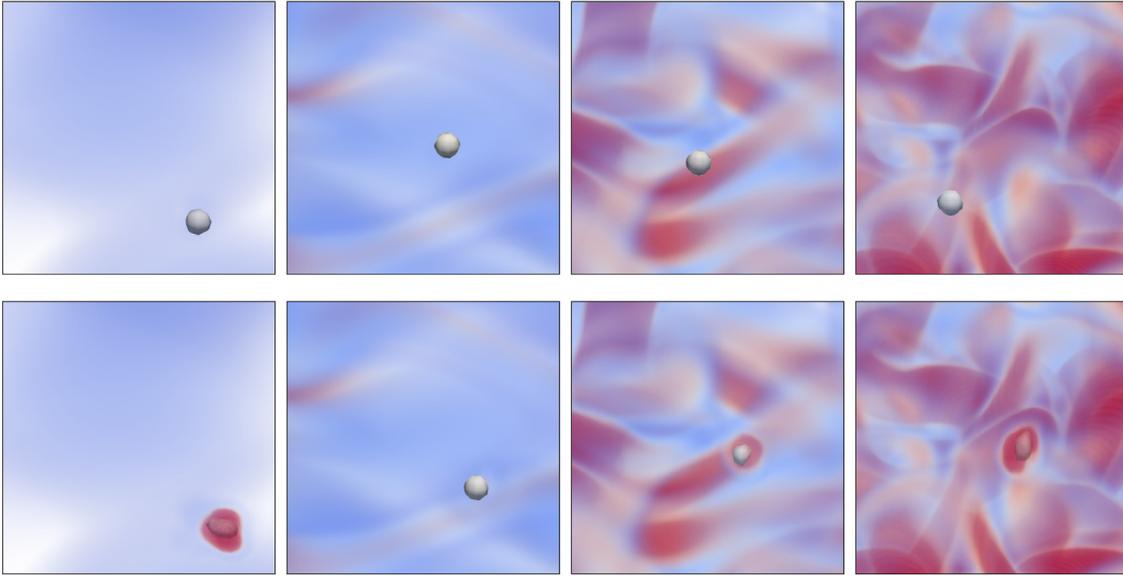
Fig. 18. Isosurfaces of the volume fraction and the magnitude of vorticity (increasing values from blue to red) with $R/h = 2.04$ at $t = 1.0$, 3.5, 6.0 and 8.5. At such resolution GHF (bottom) produces a solution with stronger spurious flow near the bubble than our method (top). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
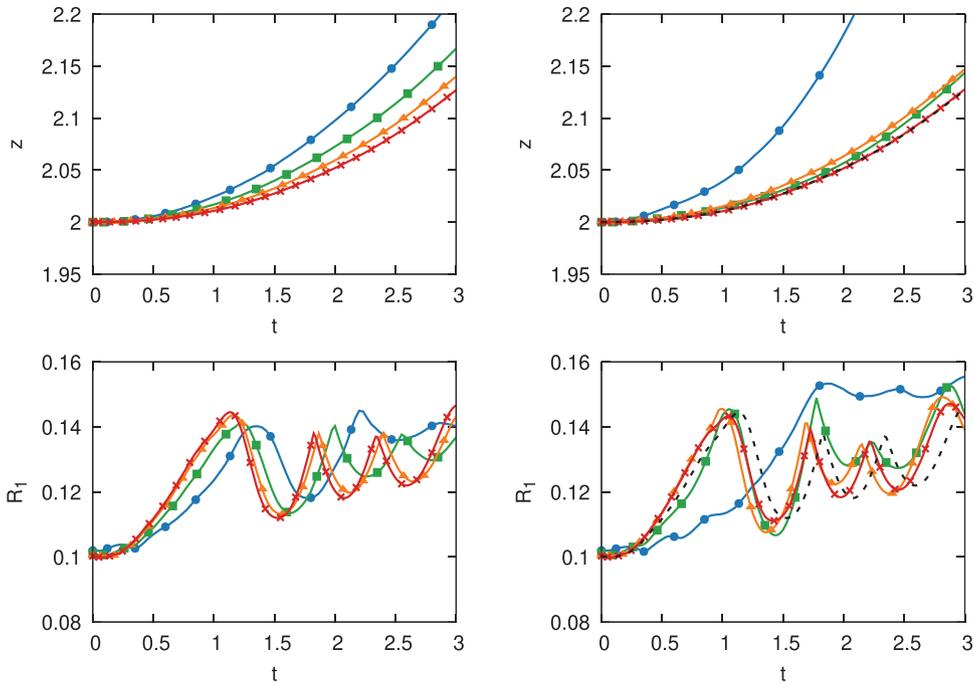
Fig. 19. Trajectory of the droplet center of mass in the z-direction and the smallest semi-axis of the ellipsoid of gyration at various resolutions $R/h = 1.53$ ⎯●⎯, 2.04 ⎯■⎯, 3.06 ⎯▲⎯ and 4.08 ⎯✕⎯: present (left) and GHF (right). The dashed line ⎯ ⎯ ⎯ on the right shows the results at the finest resolution $R/h = 4.08$ by the present method.

droplet with $R = 0.1$

$$\left(\frac{x-2}{R}\right)^2 + \left(\frac{y-2}{R}\right)^2 + \left(\frac{z-2}{5R}\right)^2 = 1. \tag{23}$$

The density and viscosity ratios are set to $\rho_2/\rho_1 = 10$ and $\mu_2/\mu_1 = 10$. Other parameters of the problem are kept the same: the Reynolds number $\mathrm{Re} = \rho_1/\mu_1 = 800$ and the Weber number $\mathrm{We} = 2\rho_1 R/\sigma = 2$.

The deformation of the droplet is characterized by the gyration tensor

$$\frac{1}{V}\int (\boldsymbol{x} - \boldsymbol{x}_c) \otimes (\boldsymbol{x} - \boldsymbol{x}_c)\,\alpha\,dV, \tag{24}$$

where $V = \int \alpha\,dV$ and $\boldsymbol{x}_c = \frac{1}{V}\int \boldsymbol{x}\alpha\,dV$. For an elliptical droplet, the principal components $\lambda_1 < \lambda_2 < \lambda_3$ of the gyration tensor relate to semi-axes $R_1 < R_2 < R_3$ of the ellipsoid as $R_i = \sqrt{5\lambda_i}$, $i = 1, 2, 3$. Fig. 19 shows the trajectory of the droplet center of mass in the z-direction and the smallest semi-axis R_1 of the ellipsoid of gyration at various resolutions in comparison to GHF. Snapshots of the vorticity magnitude and the bubble shapes are shown in Figs. 20 and 21 for both methods at resolutions $R/h = 1.53$ and 4.08. Both methods converge to the same solution with the mesh refinement. The oscillation of the droplet is indicated by $R_1(t)$ which starts at $R_1 = 0.1$ according the initial conditions and reaches the first max-
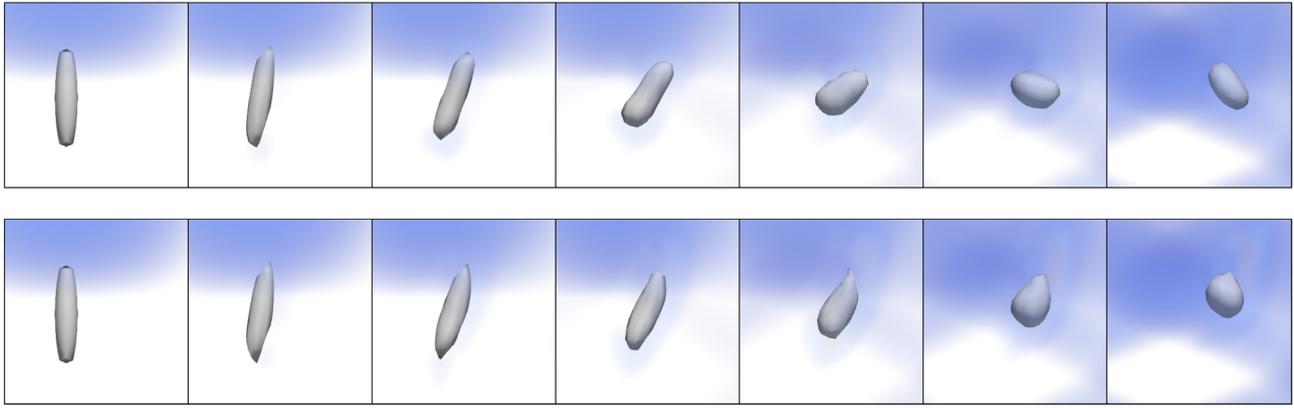
Fig. 20. Isosurfaces of the volume fraction and the magnitude of vorticity (increasing values from blue to red) with resolution $R/h = 1.53$ at $t =$ 0.0, 0.3, 0.6, 0.9, 1.2, 1.5 and 1.8: present method (top) and GHF (bottom). Unlike GHF, our method captures the oblate shape at $t = 1.5$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
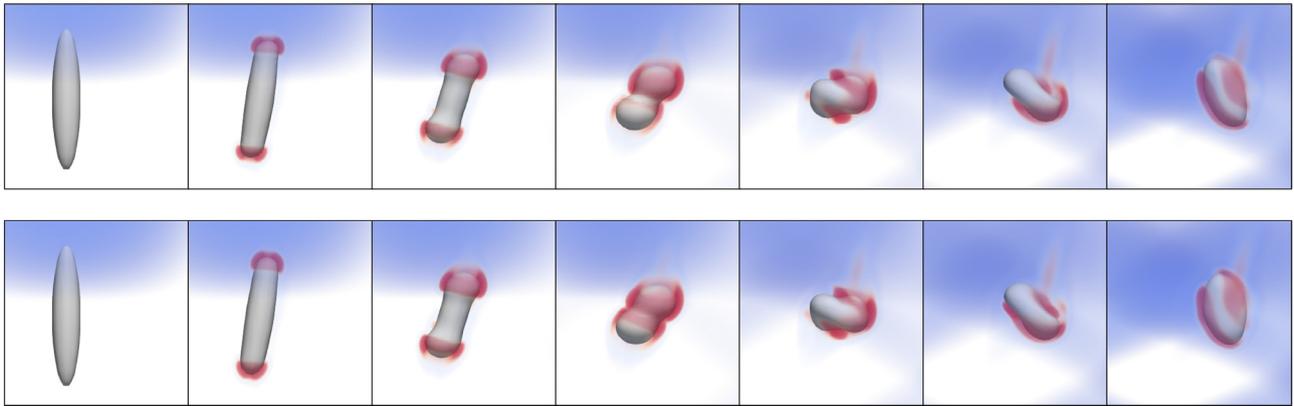
Fig. 21. Isosurfaces of the volume fraction and the magnitude of vorticity (increasing values from blue to red) with resolution $R/h = 4.08$ at $t =$ 0.0, 0.3, 0.6, 0.9, 1.2, 1.5 and 1.8: present method (top) and GHF (bottom). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

imum at $t = 1.2$ when the droplet approaches a spherical shape shown in Fig. 21. Due to inertia, the droplet further deforms to an oblate shape corresponding to a minimum of R_1 at $t = 1.5$. At the lowest resolution $R/h = 1.53$, our method captures the formation of the oblate shape at $t = 1.8$ which is delayed compared to the finest resolution. However, with GHF the oblate shape is not captured. At the next resolution $R/h = 2.04$, GHF computes the trajectory more accurately than our method.

4.3. Coalescence of bubbles

Coalescence of bubbles and drops is an actively studied phenomenon commonly found in nature and industry (Anthony et al., 2017; Soto et al., 2018). 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

$$\mathrm{Oh} = \frac{\mu_1}{\sqrt{\rho_1 R \sigma}}, \tag{25}$$

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

We refer to Soto et al. (2018) 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. Figs. 22 and 23 show the isosurfaces of the volume fraction with different resolutions at $t/T = 0.08$ and 0.18, where $T = \sqrt{\rho_1 R^3/\sigma}$ is the capillary time.
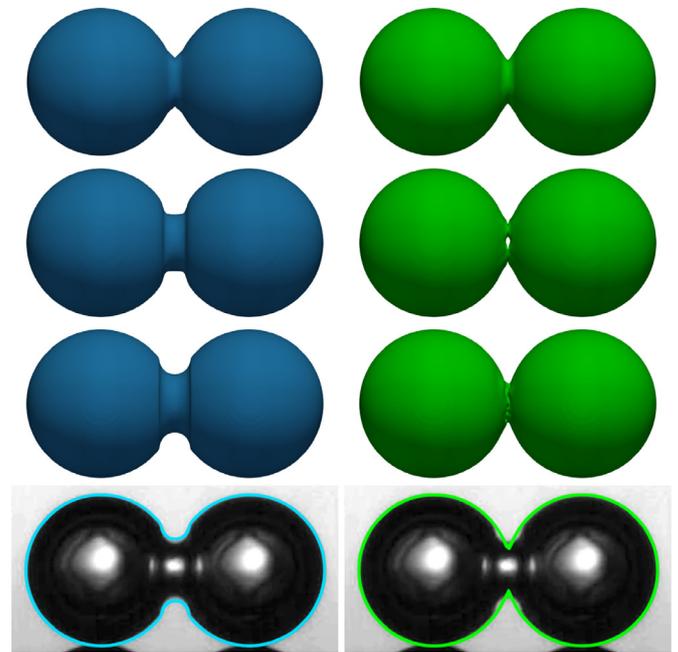
Fig. 22. Isosurfaces of volume fraction at $t/T = 0.08$ depending on resolution $R/h =$ 9.6 (top), 19.2 (middle) and 38.4 (bottom): present (left) and GHF (right). Experimental image (Soto et al., 2018) overlapped with contours from the finest mesh of each method. The time value for the experimental image is multiplied by 1.2.
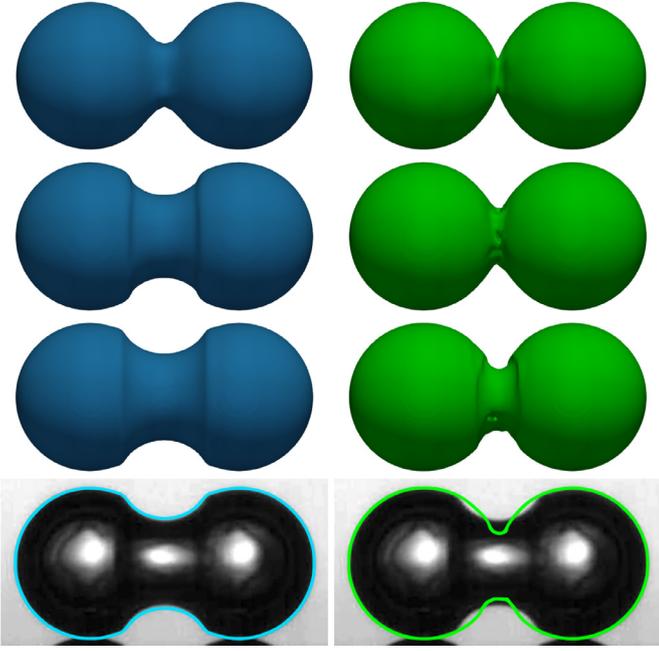
Fig. 23. Isosurfaces of volume fraction at $t/T = 0.18$ depending on resolution $R/h = 9.6$ (top), 19.2 (middle) and 38.4 (bottom): present (left) and GHF (right). Experimental image (Soto et al., 2018) overlapped with contours from the finest mesh of each method. The time value for the experimental image is multiplied by 1.2.

We found that shapes from our simulations match well the experimental results (see Figs. 22, 23 and 29) when a factor of 1.2 is applied to the values of time reported in Soto et al. (2018). We note that the simulations using the boundary integral method (Soto et al., 2018) capture the shapes reported in experiments without mentioning any such factor. We performed additional simulations using Gerris (Popinet, 2009) (see Fig. 31) and again matching shapes obtained by these simulations and those obtained experimentally required the adjustment by the factor of 1.2, in agreement with Basilisk and our own software. This discrepancy may be attributed to the use of an inviscid boundary integral method in Soto et al. (2018) and the viscous simulations employed in Gerris, Basilisk and our own software. Furthermore, we match the results of another experimental study (Thoroddsen et al., 2005) without scaling of time as shown in Section A.2 and Figures 32 and 33.

The evolution of the neck radius is presented in Fig. 24 in comparison to the experimental data reported in Soto et al. (2018). We observe that the present method agrees well with experimental data while this is not the case for simulations using the generalized height function method, in particular at later times (see Fig. 23). The generalized height-function method introduces spuri-
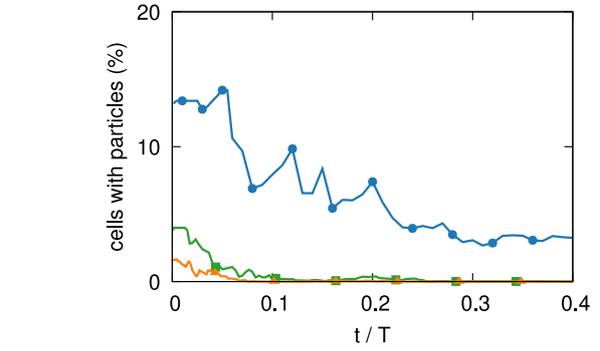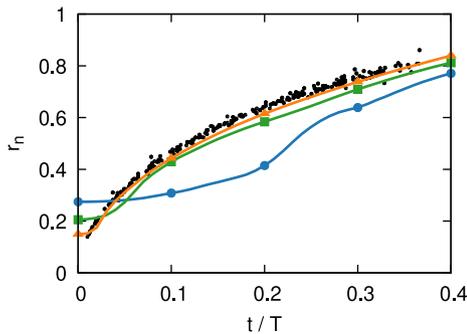
Fig. 25. Percentage of interfacial cells where the curvature is computed with particles for $R/h = 9.6$ ——●——, 19.2 ——■——, 38.4 ——▲——.
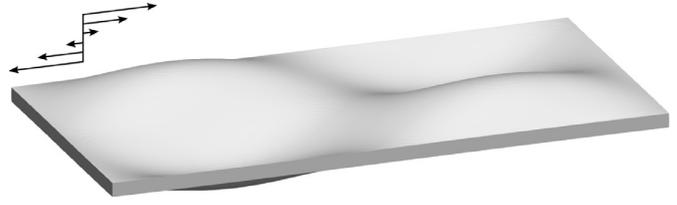
Fig. 26. Initial volume fraction representing an air sheet. Arrows indicate the direction of the shear.

ous disturbances of the interface near the coalescence neck. Furthermore, its solution does not converge with mesh refinement. The reason for this is that the initial shape has effectively infinite curvature at the coalescence neck as refining the mesh increases the curvature resolved on the mesh. As seen from Fig. 25, increasing the resolution reduces the percentage of cells where height functions are not defined and the curvature is estimated using particles. Nevertheless, more accurate estimation of curvature in such cells benefits the overall accuracy.

4.4. Breakup of air sheet in shear flow

The following test case demonstrates the deformation and breakup of an air sheet in shear flow. The domain $[0, 2] \times [0, 1] \times [0, 1]$ is periodic in the x- and z-directions and no-slip conditions are imposed on the boundaries in the y-direction. The initial velocity profile is simple shear $u_x = (y - 0.5)U/D$ and $u_y = u_z = 0$, where $D = 0.05$ is the thickness of the sheet and $U = 0.2$ is the velocity difference. The initial volume fraction

$$\alpha = \begin{cases} 1, & \left| 0.1 \sin(\pi x) \sin(2\pi z) + 0.5 - y \right| < \frac{D}{2} \\ 0, & \text{otherwise} \end{cases} \tag{26}$$
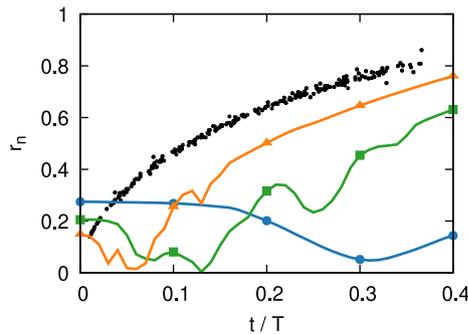
Fig. 24. Evolution of the coalescence neck radius relative to the bubble radius depending on the resolution $R/h = 9.6$ ——●——, 19.2 ——■——, 38.4 ——▲——: present (left) and GHF (right) compared to experiment (Soto et al., 2018) (dots).
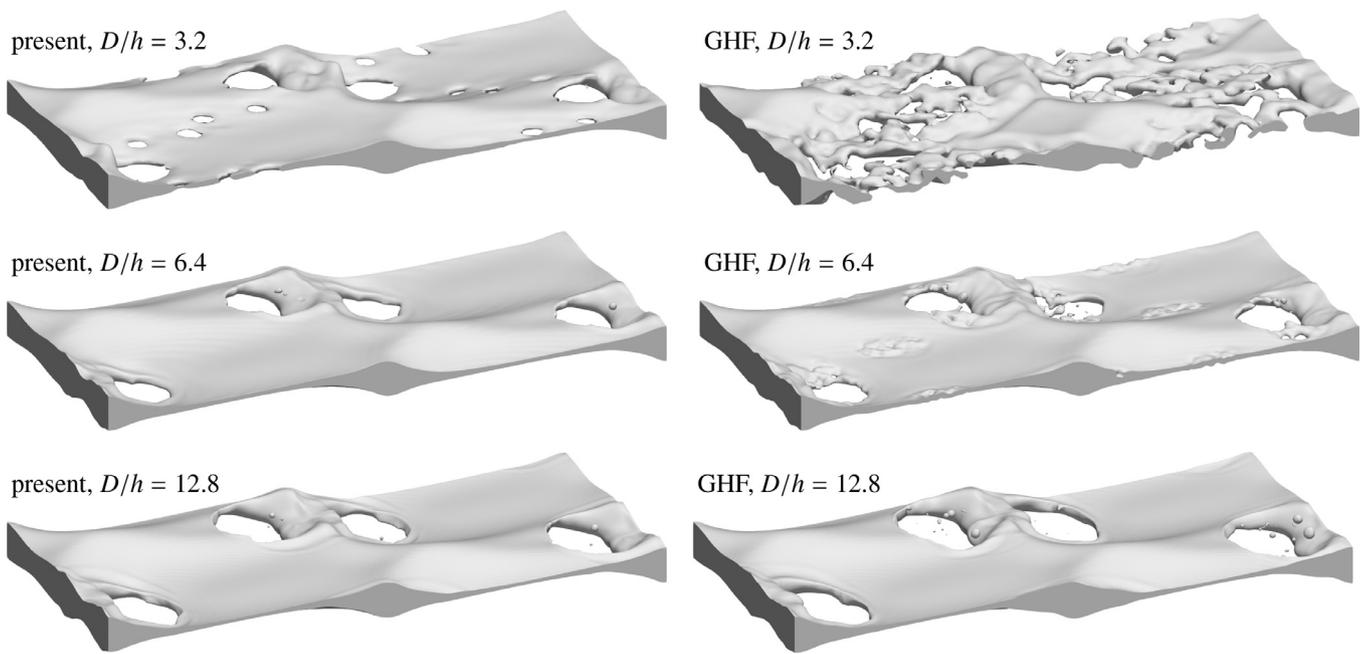
Fig. 27. Shapes of the air sheet at $t = 1.7$ computed with present method (left) and GHF (right) at resolutions D/h =3.2 (top), 6.4 (middle) and 12.8 (bottom).

is shown in Fig. 26. Parameters of the problem are the Reynolds number Re $= \rho_1 U D / \mu_1$ and the Weber number We $= \rho_1 U^2 D / \sigma$. Here we choose Re $= 80$ and We $= 0.64$. The density and viscosity ratios are set to $\rho_2/\rho_1 = 0.01$ and $\mu_2/\mu_1 = 0.01$.

Initial deformations of the air sheet develop further in the shear flow. This leads to tearing of the interface starting in four distinct locations. Fig. 27 shows the shapes computed with both methods of curvature estimation at various resolutions. Both methods converge to the same solution with mesh refinement. At lower resolutions, GHF shows more tearing of the interface, while with the present method the interface remains stable.

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 (Popinet, 2009) implemented in Basilisk. The present method is more accurate at resolutions up to four cells per curvature radius and even with one cell per radius provides the relative curvature error below 10%. 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 (Hashemi et al., 2019) and a plunging jet with air entrainment (Karnakov et al., 2019).

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 (Owkes and Desjardins, 2015). Fi-

nally, the property of recovering the exact curvature mentioned in Section A.1 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.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Petr Karnakov: Conceptualization, Methodology, Investigation, Software, Writing - original draft, Visualization. **Sergey Litvinov:** Conceptualization, Software, Validation, Writing - review & editing. **Petros Koumoutsakos:** Conceptualization, Funding acquisition, Supervision, Writing - review & editing.

Acknowledgements

Appendix A

A1. Attraction to circular arcs

The attraction force (7) can be modified to include a dependency on the current estimate of curvature. One such modification consists in replacing the line segments with circular arcs. This formulation recovers the exact curvature if the endpoints of all line segments belong to a circle. To define the force at position $\boldsymbol{x}$, we find the nearest point $\boldsymbol{y} \in L$ on the interface and the corresponding line segment $[\boldsymbol{a}_l, \boldsymbol{b}_l]$. Then we find a factor δ such that

$$\boldsymbol{x}_L(\boldsymbol{x}, \kappa) = \boldsymbol{y} + \delta \boldsymbol{n}_l \qquad (27)$$
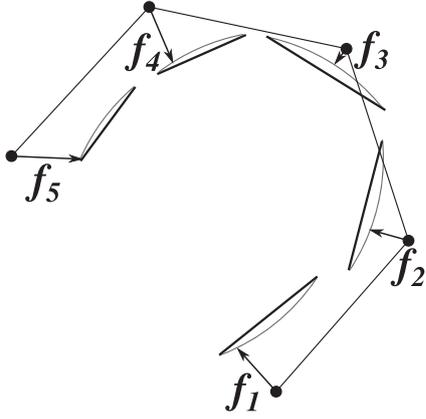
Fig. 28. Line segments represent the interface and circular arcs of curvature κ pass through their endpoints. The force acts on each particle towards the circular arc from the nearest line segment.

Fig. 30. Evolution of the coalescence neck radius relative to the bubble radius depending on the resolution $R/h = 9.6$ ●—●, 19.2 ■—■ and 38.4 ▲—▲ produced by Aphros compared to experiment (Soto et al., 2018) (dots).
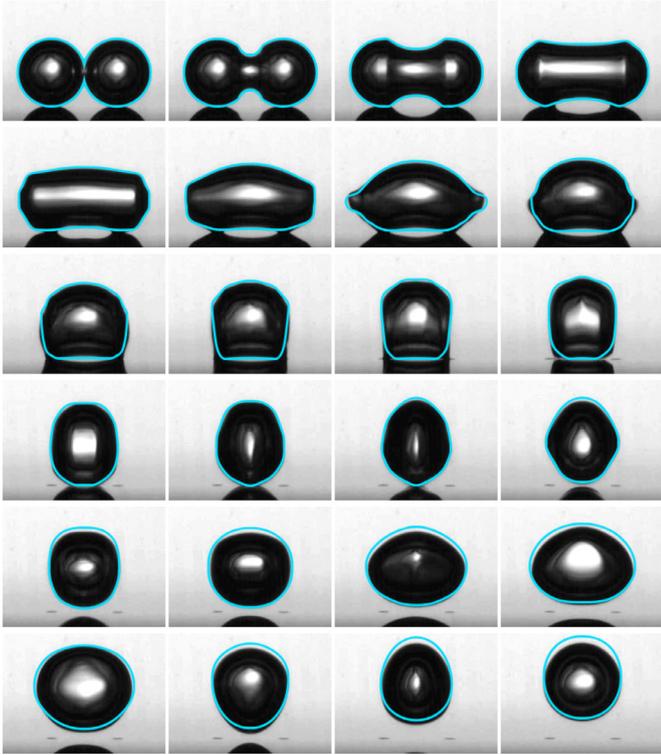
Fig. 29. Snapshots from experiment (Soto et al., 2018) overlaid by the projections of the shapes from the simulation by Aphros with resolution $R/h = 76.8$ (blue lines) at times $t/T = 0$, 0.13, 0.26, 0.4, 0.53, 0.66, 0.79, 0.92, 1.1, 1.2, 1.3, 1.4, 1.6, 1.7, 1.8, 2.1, 2.4, 2.6, 2.9, 3.2, 3.7, 4.2, 4.7 and 5.3. The time values for the experimental images are multiplied by 1.2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

belongs to a circular arc of curvature κ through the endpoints $\boldsymbol{a}_l$ and $\boldsymbol{b}_l$, where κ is a known estimation of curvature and $\boldsymbol{n}_l$ is the unit normal of $[\boldsymbol{a}_l, \boldsymbol{b}_l]$. Such δ is given by

$$\delta = \sqrt{\frac{1}{\kappa^2} - d^2} - \sqrt{\frac{1}{\kappa^2} - w^2} = \frac{\kappa(w^2 - d^2)}{\sqrt{1 - \kappa^2 d^2} + \sqrt{1 - \kappa^2 w^2}}, \qquad (28)$$

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

$$\boldsymbol{f}(\boldsymbol{x}, \kappa) = \eta \left(\boldsymbol{x}_L(\boldsymbol{x}, \kappa) - \boldsymbol{x} \right), \qquad (29)$$

where $\eta \in [0, 1]$ is a relaxation parameter. Fig. 28 illustrates the computation of forces after replacing the line segments with circular arcs.
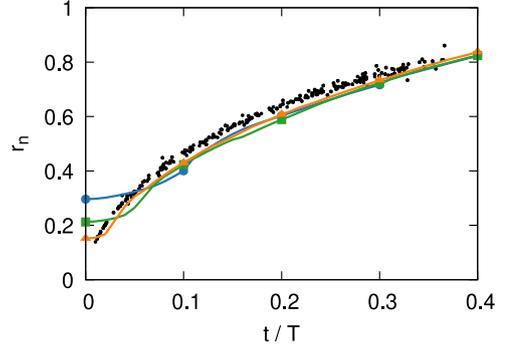
A2. Coalescence of bubbles with SIMPLE-based solver

The method for curvature estimation is also implemented as part of our in-house multiphase flow solver Aphros. We use a finite volume discretization based on the SIMPLE method for the pressure coupling (Patankar and Spalding, 1983; Ferziger and Peric, 2012) and the second-order scheme QUICK (Leonard, 1979) for convective fluxes. The advection equation is solved using the volume-of-fluid method PLIC with piecewise linear reconstruction (Aulisa et al., 2007), 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 (Francois et al., 2006) (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 (Wermelinger et al., 2018; Cubism), an open-source C++ framework for distributed parallel solvers on structured grids. To solve the linear systems, we use the GMRES method (Saad and Schultz, 1986) for the momentum equation and the preconditioned conjugate gradient method (Ashby and Falgout, 1996) for the pressure correction implemented in the Hypre library (Falgout and Yang, 2002; HYPRE).

One functionality of our code is the ability to describe static contact angles of 0 and 180 degrees. This feature is implemented using ghost cells. Assuming that $\alpha = 1$ is the volume fraction inside the bubble or droplet, we fill two layers of ghost cells adjacent to boundaries with values $\alpha = 0$ and $\alpha = 1$ for the contact angles of 180 and 0 degrees respectively. Based on such volume fraction field, we estimate the normals and curvature in all domain cells with the usual algorithm. This allows us to describe the evolution of the two bubbles after coalescence presented in Section 4.3 at later stages as well and include the detachment from the solid wall and oscillations. The problem formulation is closer to the experimental conditions than in Section 4.3. The bubbles are initially placed near a solid wall with the imposed contact angle of 180 degrees and the gravitational acceleration is considered with the Eötvös number of Eo $= \rho g R^2/\sigma = 0.061$. Results of the simulation compared to the experimental images are shown in Fig. 29.

Furthermore, the evolution of the coalescence neck radius on the coarsest resolution is more accurate than in Basilisk as shown in Figs. 24 and 30. This is attributed to differences in the algorithm for computation of interface normals and the way the estimates of curvature are transferred from cells to faces (average over neighbors in Basilisk, and the cell with the minimal $|\alpha - \frac{1}{2}|$ in Aphros).
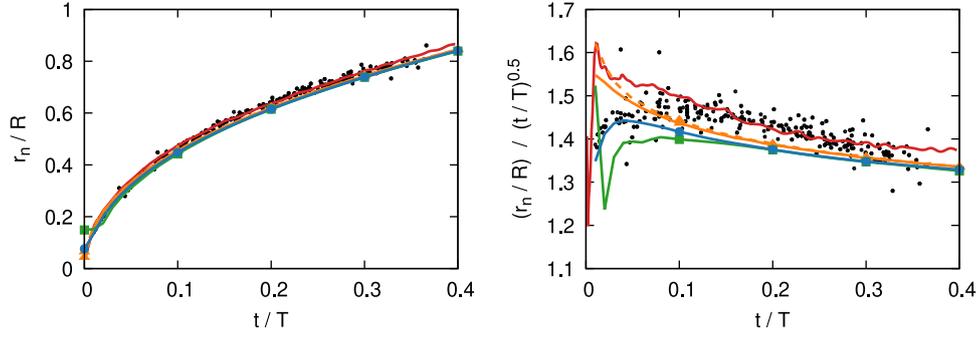
Fig. 31. Evolution of the coalescence neck radius relative to the bubble radius (left) and the coalescence neck radius divided by $\sqrt{t}$ (right). Gerris in the axisymmetric formulation with adaptive mesh refinement with the equivalent resolution $R/h = 307.2$ - - ▲ - - and $R/h = 614.4$ ——▲——, Basilisk with proposed method for curvature at $R/h = 38.4$ ——■—— and Aphros at $R/h = 153.6$ ——●——. Experiment (Soto et al., 2018) (dots) and boundary integral (Soto et al., 2018) ———— without scaling of time.

$\frac{\mu_1}{\sqrt{\rho_1 R \sigma}}$ is set to Oh $= 0.007$. The experimental images overlaid with the results of the simulation are shown in Fig. 32 and the evolution of coalescence neck is presented in Fig. 33. In this case, our results agree with the experimental data without scaling of time.

A3. Derivatives of particle positions

The algorithm in Section 2.2 requires derivatives of particle positions (5) with respect to angles ϕ and θ which are given as

$$\frac{\partial \boldsymbol{x}_i}{\partial \phi} = \begin{cases} \sum\limits_{j=1}^{i-c} h_p \boldsymbol{e}\big(\phi + (j - \tfrac{1}{2})\theta + \tfrac{\pi}{2}\big) & i > c, \\ \boldsymbol{0} & i = c, \\ -\sum\limits_{j=1}^{c-i} h_p \boldsymbol{e}\big(\phi - (j - \tfrac{1}{2})\theta + \tfrac{\pi}{2}\big) & i < c, \end{cases} \tag{30}$$

$$\frac{\partial \boldsymbol{x}_i}{\partial \theta} = \begin{cases} \sum\limits_{j=1}^{i-c} h_p \big(j - \tfrac{1}{2}\big) \boldsymbol{e}\big(\phi + (j - \tfrac{1}{2})\theta + \tfrac{\pi}{2}\big) & i > c, \\ \boldsymbol{0} & i = c, \\ \sum\limits_{j=1}^{c-i} h_p \big(j - \tfrac{1}{2}\big) \boldsymbol{e}\big(\phi - (j - \tfrac{1}{2})\theta + \tfrac{\pi}{2}\big) & i < c. \end{cases} \tag{31}$$

The following recurrence relations are useful for the implementation:

$$\boldsymbol{x}_i = \begin{cases} \boldsymbol{x}_{i-1} + h_p \boldsymbol{e}\big(\phi + (i - c - \tfrac{1}{2})\theta\big) & i > c, \\ \boldsymbol{p} & i = c, \\ \boldsymbol{x}_{i+1} - h_p \boldsymbol{e}\big(\phi - (c - i - \tfrac{1}{2})\theta\big) & i < c, \end{cases} \tag{32}$$

$$\frac{\partial \boldsymbol{x}_i}{\partial \phi} = \begin{cases} \frac{\partial \boldsymbol{x}_{i-1}}{\partial \phi} + h_p \boldsymbol{e}\big(\phi + (i - c - \tfrac{1}{2})\theta + \tfrac{\pi}{2}\big) & i > c, \\ \boldsymbol{0} & i = c, \\ \frac{\partial \boldsymbol{x}_{i+1}}{\partial \phi} - h_p \boldsymbol{e}\big(\phi - (c - i - \tfrac{1}{2})\theta + \tfrac{\pi}{2}\big) & i < c, \end{cases} \tag{33}$$

$$\frac{\partial \boldsymbol{x}_i}{\partial \theta} = \begin{cases} \frac{\partial \boldsymbol{x}_{i-1}}{\partial \theta} + h_p (i - c - \tfrac{1}{2}) \boldsymbol{e}\big(\phi + (i - c - \tfrac{1}{2})\theta + \tfrac{\pi}{2}\big) & i > c, \\ \boldsymbol{0} & i = c, \\ \frac{\partial \boldsymbol{x}_{i+1}}{\partial \theta} + h_p (c - i - \tfrac{1}{2}) \boldsymbol{e}\big(\phi - (c - i - \tfrac{1}{2})\theta + \tfrac{\pi}{2}\big) & i < c. \end{cases} \tag{34}$$

Fig. 32. Snapshots from experiment (Thoroddsen et al., 2005) overlaid by the projections of the shapes from the simulation by Aphros with resolution $R/h = 76.8$ (blue lines) at times $t/T = 0$, 0.0061, 0.012, 0.018, 0.031, 0.043, 0.061, 0.08, 0.12, 0.17, 0.21 and 0.27. Experimental images reprinted from (Thoroddsen et al., 2005), with the permission of AIP Publishing. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
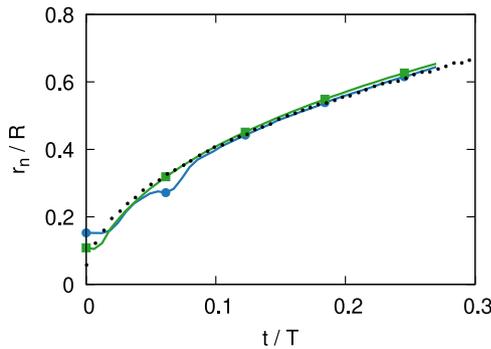
Fig. 33. Evolution of the neck radius depending on the resolution $R/h = 38.4$ ——●—— and 76.8 ——■—— produced by Aphros compared to experiment (Thoroddsen et al., 2005) (dots).

A4. Proof of convergence of iterations

We have also considered another experimental study of bubble coalescence (Thoroddsen et al., 2005). Initially, both bubbles are positioned along the z-axis and have elliptical shapes. One bubble with semi-axes $R_x = R_y = 1.08 R$ and $R_z = 1.0 R$ is positioned above the other bubble with semi-axes $R_x = R_y = 0.92 R$ and $R_z = 0.97 R$, where R is their characteristic size. The Ohnesorge number Oh $=$

We show that iterations in Section 2.2 can be formulated as steps of the gradient descent minimizing an energy function. Given the initial conditions, the central particle $\boldsymbol{x}_c$ already belongs to a line segment. Therefore, the corresponding force is zero $\boldsymbol{f}_c^m = 0$, and Step 1 trivializes such that $\boldsymbol{p}^m = \boldsymbol{p}^0$, $\boldsymbol{X}^* = \boldsymbol{X}^m$ and $\boldsymbol{F}^* = \boldsymbol{F}^m$. In

the following, the argument $\boldsymbol{p}$ is omitted. We define the energy in terms of the remaining parameters ϕ and θ

$$E(\phi, \theta) = \frac{1}{2} \sum_{i=1}^{N} \min_{\boldsymbol{y} \in L} |\boldsymbol{y} - \boldsymbol{x}_i(\phi, \theta)|^2. \tag{35}$$

The energy can be expressed as a superposition

$$E(\phi, \theta) = \hat{E}(\phi, \theta, \boldsymbol{Y}_{\min}(\phi, \theta)) \tag{36}$$

of two functions

$$\hat{E}(\phi, \theta, \boldsymbol{Y}) = \frac{1}{2} \sum_{i=1}^{N} |\boldsymbol{y}_i - \boldsymbol{x}_i(\phi, \theta)|^2 \tag{37}$$

$$\boldsymbol{Y}_{\min}(\phi, \theta) = [\arg\min_{\boldsymbol{y} \in L} |\boldsymbol{y} - \boldsymbol{x}_i(\phi, \theta)|, \; i = 1, \ldots, N]. \tag{38}$$

Derivatives of $\hat{E}$ read

$$\frac{\partial \hat{E}}{\partial \phi}(\phi, \theta, \boldsymbol{Y}) = -(\boldsymbol{Y} - \boldsymbol{X}(\phi, \theta)) \cdot \frac{\partial \boldsymbol{X}}{\partial \phi}(\phi, \theta)$$

$$\frac{\partial \hat{E}}{\partial \theta}(\phi, \theta, \boldsymbol{Y}) = -(\boldsymbol{Y} - \boldsymbol{X}(\phi, \theta)) \cdot \frac{\partial \boldsymbol{X}}{\partial \theta}(\phi, \theta).$$

In this notation, the force defined by (7) and (9) transforms to

$$\boldsymbol{F}(\phi, \theta) = \eta \left(\boldsymbol{Y}_{\min}(\phi, \theta) - \boldsymbol{X}(\phi, \theta) \right). \tag{39}$$

Taking into account

$$\boldsymbol{F}^* \cdot \boldsymbol{D}_\phi = -\eta \frac{\partial \hat{E}}{\partial \phi}(\phi^m, \theta^m, \boldsymbol{Y}^m)$$

$$\boldsymbol{F}^* \cdot \boldsymbol{D}_\theta = -\eta \left[\frac{\partial \hat{E}}{\partial \theta}(\phi^{m+1}, \theta^m, \boldsymbol{Y}^m) - \left(\boldsymbol{X}(\phi^{m+1}, \theta^m) - \boldsymbol{X}(\phi^m, \theta^m) \right) \right],$$

corrections in Step 2 and Step 3 can be expressed as

$$\phi^{m+1} = \phi^m - \frac{\eta}{\|\boldsymbol{D}_\phi\|^2} \frac{\partial \hat{E}}{\partial \phi}(\phi^m, \theta^m, \boldsymbol{Y}^m) \tag{40}$$

$$\theta^{m+1} = \theta^m - \frac{\eta}{\|\boldsymbol{D}_\theta\|^2} \frac{\partial \hat{E}}{\partial \theta}(\phi^{m+1}, \theta^m, \boldsymbol{Y}^m) - \tag{41}$$

$$- \frac{1 - \eta}{\|\boldsymbol{D}_\theta\|^2} \left(\boldsymbol{X}(\phi^{m+1}, \theta^m) - \boldsymbol{X}(\phi^m, \theta^m) \right) \cdot \boldsymbol{D}_\theta,$$

where $\boldsymbol{F}^* = \boldsymbol{F}(\phi^m, \theta^m)$, $\boldsymbol{D}_\phi = \frac{\partial \boldsymbol{X}}{\partial \phi}(\phi^m, \theta^m)$, $\boldsymbol{D}_\theta = \frac{\partial \boldsymbol{X}}{\partial \theta}(\phi^{m+1}, \theta^m)$ and $\boldsymbol{Y}^m = \boldsymbol{Y}_{\min}(\phi^m, \theta^m)$. The second correction can be rewritten

$$\theta^{m+1} = \theta^m - \frac{\eta}{\|\boldsymbol{D}_\phi\|^2} \frac{\partial \hat{E}}{\partial \theta}(\phi^{m+1}, \theta^m, \boldsymbol{Y}^m) + \tag{42}$$

$$+ \frac{(1 - \eta)\eta \, \tilde{\boldsymbol{D}}_\phi \cdot \boldsymbol{D}_\theta}{\|\boldsymbol{D}_\phi\|^2 \|\boldsymbol{D}_\theta\|^2} \frac{\partial \hat{E}}{\partial \phi}(\phi^m, \theta^m, \boldsymbol{Y}^m)$$

in terms of the finite difference

$$\tilde{\boldsymbol{D}}_\phi = \frac{\boldsymbol{X}(\phi^{m+1}, \theta^m) - \boldsymbol{X}(\phi^m, \theta^m)}{\phi^{m+1} - \phi^m}. \tag{43}$$

Function $\hat{E}(\phi, \theta, \boldsymbol{Y})$ is smooth on a compact set $[0, 2\pi] \times [0, 2\pi] \times L^N$, and therefore its gradient is Lipschitz continuous. In particular, for a constant $\lambda > 0$

$$\begin{aligned} |\frac{\partial \hat{E}}{\partial \phi}(\phi + \Delta\phi, \theta, \boldsymbol{Y}) - \frac{\partial \hat{E}}{\partial \phi}(\phi, \theta, \boldsymbol{Y})| &\leq \lambda \, \Delta\phi, \\ |\frac{\partial \hat{E}}{\partial \theta}(\phi, \theta + \Delta\theta, \boldsymbol{Y}) - \frac{\partial \hat{E}}{\partial \theta}(\phi, \theta, \boldsymbol{Y})| &\leq \lambda \, \Delta\theta. \end{aligned} \tag{44}$$

This implies that

$$|\Delta E - (E_\phi \Delta\phi + E_\theta \Delta\theta)| \leq \lambda (\Delta\phi^2 + \Delta\theta^2), \tag{45}$$

where $\Delta\phi = \phi^{m+1} - \phi^m$, $\Delta\theta = \theta^{m+1} - \theta^m$, $\Delta E = \hat{E}(\phi^{m+1}, \theta^{m+1}, \boldsymbol{Y}^m) - \hat{E}(\phi^m, \theta^m, \boldsymbol{Y}^m)$, $E_\phi = \frac{\partial \hat{E}}{\partial \phi}(\phi^m, \theta^m, \boldsymbol{Y}^m)$ and $E_\theta = \frac{\partial \hat{E}}{\partial \theta}(\phi^{m+1}, \theta^m, \boldsymbol{Y}^m)$. Therefore, the change of the energy after both corrections is bounded as

$$\Delta E \leq E_\phi \Delta\phi + E_\theta \Delta\theta + \lambda(\Delta\phi^2 + \Delta\theta^2). \tag{46}$$

We show that for a sufficiently small $\eta > 0$,

$$E_\phi \Delta\phi + E_\theta \Delta\theta + \lambda(\Delta\phi^2 + \Delta\theta^2) \leq 0, \tag{47}$$

which, given (40) and (42), is equivalent to

$$-\boldsymbol{v}^T \boldsymbol{A} \boldsymbol{v} + \lambda\eta \, |\boldsymbol{A}\boldsymbol{v}|^2 \leq 0, \tag{48}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \frac{1}{\|\boldsymbol{D}_\phi\|^2} & 0 \\ -\frac{(1-\eta)\, \tilde{\boldsymbol{D}}_\phi \cdot \boldsymbol{D}_\theta}{\|\boldsymbol{D}_\phi\|^2 \|\boldsymbol{D}_\theta\|^2} & \frac{1}{\|\boldsymbol{D}_\theta\|^2} \end{bmatrix} \quad \boldsymbol{v} = \begin{bmatrix} E_\phi \\ E_\theta \end{bmatrix}. \tag{49}$$

This follows from a stronger inequality

$$-\boldsymbol{v}^T \boldsymbol{A} \boldsymbol{v} + \lambda\eta \, \|\boldsymbol{A}\|^2 |\boldsymbol{v}|^2 \leq 0 \tag{50}$$

or, in terms of a symmetric matrix,

$$\boldsymbol{v}^T \left(\frac{\boldsymbol{A} + \boldsymbol{A}^T}{2} - \lambda\eta \, \|\boldsymbol{A}\|^2 \boldsymbol{I} \right) \boldsymbol{v} \geq 0, \tag{51}$$

which states that the matrix is positive semi-definite. As seen from (30) and (31), derivatives $\boldsymbol{D}_\phi$ and $\boldsymbol{D}_\theta$ are uniformly bounded away from zero and infinity. Therefore, it is sufficient to show that matrix $\frac{\boldsymbol{A} + \boldsymbol{A}^T}{2}$ is positive definite in the limiting case $\eta \to 0$

$$\frac{\boldsymbol{A} + \boldsymbol{A}^T}{2} = \begin{bmatrix} \frac{1}{\|\boldsymbol{D}_\phi\|^2} & -\frac{1}{2} \frac{\boldsymbol{D}_\phi \cdot \boldsymbol{D}_\theta}{\|\boldsymbol{D}_\phi\|^2 \|\boldsymbol{D}_\theta\|^2} \\ -\frac{1}{2} \frac{\boldsymbol{D}_\phi \cdot \boldsymbol{D}_\theta}{\|\boldsymbol{D}_\phi\|^2 \|\boldsymbol{D}_\theta\|^2} & \frac{1}{\|\boldsymbol{D}_\theta\|^2} \end{bmatrix} \tag{52}$$

as the finite difference $\tilde{\boldsymbol{D}}_\phi$ from (43) uniformly converges to $\boldsymbol{D}_\phi$. The above is equivalent to

$$\frac{(\boldsymbol{D}_\phi \cdot \boldsymbol{D}_\theta)^2}{\|\boldsymbol{D}_\phi\|^2 \|\boldsymbol{D}_\theta\|^2} < 4, \tag{53}$$

which always holds for a scalar product. This shows that corrections of parameters ϕ and θ reduce the energy

$$\hat{E}(\phi^{m+1}, \theta^{m+1}, \boldsymbol{Y}^m) \leq \hat{E}(\phi^m, \theta^m, \boldsymbol{Y}^m). \tag{54}$$

Finally, definition (38) implies that

$$\hat{E}(\phi^{m+1}, \theta^{m+1}, \boldsymbol{Y}_{\min}(\phi^{m+1}, \theta^{m+1})) \leq \hat{E}(\phi^{m+1}, \theta^{m+1}, \boldsymbol{Y}^m). \tag{55}$$

Combining the last two inequalities, we arrive at

$$\hat{E}(\phi^{m+1}, \theta^{m+1}, \boldsymbol{Y}^{m+1}) \leq \hat{E}(\phi^m, \theta^m, \boldsymbol{Y}^m) \tag{56}$$

or, in the original notation (36),

$$E(\phi^{m+1}, \theta^{m+1}) \leq E(\phi^m, \theta^m). \tag{57}$$

Together with $E \geq 0$, this guarantees convergence of iterations.

Supplementary material

References

Aniszewski, W., Zaleski, S., Popinet, S., Saade, Y., 2019. Planar Jet Stripping of Liquid Coatings: Numerical Studies. arXiv preprint arXiv:1907.07659.

Anthony, C.R., Kamat, P.M., Thete, S.S., Munro, J.P., Lister, J.R., Harris, M.T., Basaran, O.A., 2017. Scaling laws and dynamics of bubble coalescence. Phys. Rev. Fluids 2 (8), 83601.

Ashby, S.F., Falgout, R.D., 1996. A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations. Nucl. Sci. Eng. 124 (1), 145–159.

Aulisa, E., Manservisi, S., Scardovelli, R., Zaleski, S., 2007. Interface reconstruction with least-squares fit and split advection in three-dimensional cartesian geometry. J. Comput. Phys. 225 (2), 2301–2319.

Basilisk, 2014. http://basilisk.fr/.

Bell, J.B., Colella, P., Glaz, H.M., 1989. A second-order projection method for the incompressible Navier-Stokes equations. J. Comput. Phys. 85 (2), 257–283.

Boissonneau, P., Byrne, P., 2000. An experimental investigation of bubble-induced free convection in a small electrochemical cell. J. Appl. Electrochem. 30 (7), 767–775.

Bornia, G., Cervone, A., Manservisi, S., Scardovelli, R., Zaleski, S., 2011. On the properties and limitations of the height function method in two-dimensional cartesian geometry. J. Comput. Phys. 230 (4), 851–862.

Brackbill, J.U., Kothe, D.B., Zemach, C., 1992. A continuum method for modeling surface tension. J. Comput. Phys. 100 (2), 335–354.

Chorin, A.J., 1968. Numerical solution of the Navier-Stokes equations. Math. Comput. 22 (104), 745–762.

Cubism, 2016. Cubism: Parallel Block Processing Library. https://gitlab.ethz.ch/mavt-cse/Cubism.

Cummins, S.J., Francois, M.M., Kothe, D.B., 2005. Estimating curvature from volume fractions. Comput. Struct. 83 (6–7), 425–434.

Diwakar, S., Das, S.K., Sundararajan, T., 2009. A quadratic spline based interface (quasi) reconstruction algorithm for accurate tracking of two-phase flows. J. Comput. Phys. 228 (24), 9107–9130.

Evrard, F., Denner, F., van Wachem, B., 2017. Estimation of curvature from volume fractions using parabolic reconstruction on two-dimensional unstructured meshes. J. Comput. Phys. 351, 271–294.

Falgout, R.D., Yang, U.M., 2002. hypre: a library of high performance preconditioners. In: International Conference on Computational Science. Springer, pp. 632–641.

Ferziger, J.H., Peric, M., 2012. Computational Methods for Fluid Dynamics. Springer Science & Business Media.

Francois, M.M., Cummins, S.J., Dendy, E.D., Kothe, D.B., Sicilian, J.M., Williams, M.W., 2006. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. J. Comput. Phys. 213 (1), 141–173.

Hashemi, S.M.H., Karnakov, P., Hadikhani, P., Chinello, E., Litvinov, S., Moser, C., Koumoutsakos, P., Psaltis, D., 2019. A versatile and membrane-less electrochemical reactor for the electrolysis of water and brine. Energy Environ. Sci..

HYPRE: Scalable Linear Solvers, 2005. https://computation.llnl.gov/projects/hypre-scalable-linear-solvers-multigrid-methods.

Jakobsen, H.A., Lindborg, H., Dorao, C.A., 2005. Modeling of bubble column reactors: progress and limitations. Ind. Eng. Chem. Res. 44 (14), 5107–5151.

Karnakov, P., Wermelinger, F., Chatzimanolakis, M., Litvinov, S., Koumoutsakos, P., 2019. A high performance computing framework for multiphase, turbulent flows on structured grids. In: Proceedings of the Platform for Advanced Scientific Computing Conference. ACM.

Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: active contour models. Int. J. Comput. Vis. 1 (4), 321–331.

Kharangate, C.R., Mudawar, I., 2017. Review of computational studies on boiling and condensation. Int. J. Heat Mass Transf. 108, 1164–1196.

Kiger, K.T., Duncan, J.H., 2012. Air-entrainment mechanisms in plunging jets and breaking waves. Annu. Rev. Fluid Mech. 44, 563–596.

Leonard, B.P., 1979. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. Comput. Methods Appl. Mech. Eng. 19 (1), 59–98.

Ling, Y., Zaleski, S., Scardovelli, R., 2015. Multiscale simulation of atomization with small droplets represented by a lagrangian point-particle model. Int. J. Multiph. Flow 76, 122–143.

Owkes, M., Desjardins, O., 2015. A mesh-decoupled height function method for computing interface curvature. J. Comput. Phys. 281, 285–300.

Patankar, S.V., Spalding, D.B., 1983. 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, pp. 54–73.

Popinet, S., 2009. An accurate adaptive solver for surface-tension-driven interfacial flows. J. Comput. Phys. 228 (16), 5838–5866.

Popinet, S., 2018. Numerical models of surface tension. Annu. Rev. Fluid Mech. 50, 49–75.

Renardy, Y., Renardy, M., 2002. Prost: a parabolic reconstruction of surface tension for the volume-of-fluid method. J. Comput. Phys. 183 (2), 400–421.

Saad, Y., Schultz, M.H., 1986. Gmres: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat.Comput. 7 (3), 856–869.

Scardovelli, R., Zaleski, S., 1999. Direct numerical simulation of free-surface and interfacial flow. Annu. Rev. Fluid Mech. 31 (1), 567–603.

Scardovelli, R., Zaleski, S., 2000. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. J. Comput. Phys. 164 (1), 228–237.

Soto, Á., Maddalena, T., Fraters, A., Van Der Meer, D., Lohse, D., 2018. Coalescence of diffusively growing gas bubbles. J Fluid Mech 846, 143–165.

Strobl, S., Formella, A., Pöschel, T., 2016. Exact calculation of the overlap volume of spheres and mesh elements. J. Comput. Phys. 311, 158–172.

Sussman, M., Fatemi, E., Smereka, P., Osher, S., 1998. An improved level set method for incompressible two-phase flows. Comput. Fluids 27 (5–6), 663–680.

Sussman, M., Ohta, M., 2006. High-order techniques for calculating surface tension forces. In: Free Boundary Problems. Springer, pp. 425–434.

Thoroddsen, S., Etoh, T., Takehara, K., Ootsuka, N., 2005. On the coalescence speed of bubbles. Phys. Fluids 17 (7), 71703.

Van Rees, W.M., Leonard, A., Pullin, D., Koumoutsakos, P., 2011. A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers. J. Comput. Phys. 230 (8), 2794–2805.

Wenzel, E.A., Garrick, S.C., 2018. Finite particle methods for computing interfacial curvature in volume of fluid simulations. Atom. Sprays 28 (2).

Wermelinger, F., Rasthofer, U., Hadjidoukas, P.E., Koumoutsakos, P., 2018. Petascale simulations of compressible flows with interfaces. J. Comput. Sci. 26, 217–225.

Weymouth, G.D., Yue, D.K.-P., 2010. Conservative volume-of-fluid method for free–surface simulations on cartesian-grids. J. Comput. Phys. 229 (8), 2853–2865.