fluids

MDPI

Article

Evaluation and Improvements to Interfacial Curvature Predictions in `interFoam`

Arpit Agarwal [†], **Mohan Ananth and Mario F. Trujillo** [*,†]

Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA; agarwal@wisc.edu (A.A.); mananth2@wisc.edu (M.A.)
* Correspondence: mtrujillo@wisc.edu
† These authors contributed equally to this work.

Abstract: Improvements to the interfacial curvature of `interFoam` based on (i) the smoothing of the liquid fraction field and (ii) the creation of a signed distance function (ϕ-based) are implemented. While previous work in this area has focused on evaluating spurious currents and similar configurations, the tests implemented in this work are more applicable to sprays and hydrodynamic breakup problems. For the ϕ-based method, a dual approach is developed based on a geometric reconstruction of the interface at interfacial cells and the solution of the Hamilton-Jacobi equation away from these cells. The more promising results are from this method, where the lack of convergence of Laplace pressure predictions existing in the standard version of `interFoam` is fixed, resulting in second-order convergence. Similar but less drastic improvements are observed for other exercises consisting of the oscillation of a droplet, a 2-phase Orr–Sommerfeld problem, the Rayleigh–Plateau instability, and the retraction of a liquid column. It is only when the dynamics are either entirely governed by surface tension or are heavily influenced by it that we see the need to substitute the standard `interFoam` curvature approach with a more accurate scheme. For more realistic problems, which naturally include more complicated dynamics, the difference between the standard approach and the ϕ-based approach is minimal.

Keywords: `interFoam`; interfacial curvature; volume-of-fluid; `OpenFoam`

check for updates

Citation: Agarwal, A.; Ananth, M.; Trujillo, M.F. Evaluation and Improvements to Interfacial Curvature Predictions in `interFoam`. *Fluids* **2022**, *7*, 128. https://doi.org/10.3390/fluids7040128

Academic Editors: Federico Piscaglia and Jérôme Hélie

Received: 29 January 2022
Accepted: 16 March 2022
Published: 1 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Over the last decade, Volume-of-Fluid (VoF) methods and their variants [1] have gained popularity in the simulation of two-phase flows due in part to their mass conservative nature. Among these methods, the algebraic VOF methodology encapsulated in `interFoam` has been available for a large number of years as part of the open source distribution of OpenFOAM and has attracted significant attention (A 10 March 2022 search in the Google Scholar database revealed 3030 sources that included the word "`interFoam`" anywhere in the article). In this method, the effects of numerical diffusion are assessed based on a compressive interface capturing methodology that was improved by Ubbink and Issa [2] and Rusche [3] with contributions from Henry Weller. More recently, new and often more accurate VoF methodologies have been implemented in OpenFOAM, for example, the isoAdvector geometric VoF with reconstructed distance function [4,5], the addition of the geometric level set [6], the improvement to volume fraction (α) advection in complex mesh topologies [7], the implementation of a coupled level set VoF (CLSVOF) [8], and the introduction of a capillary pressure jump modeling [9], among other contributions.

Despite the availability of other VoF methodologies within OpenFOAM, or even other open source frameworks [10], `interFoam` continues to be a popular choice. It can be speculated that part of the reason for its continued use is the familiarity of the solver among its many users as well as the development of submodels within its infrastructure, which can pose a significant time investment for their implementation in other solvers. Regardless of the reason for its continued use, `interFoam` has its weaknesses, with the most prominent being arguably the calculation of interfacial curvature, κ. This has been known for some time [11] and has spawned various flavors of numerical fixes.

It is well established [1] that the major difficulty with a curvature calculation within a VoF methodology is the high degree of error associated with the calculation of second-order derivatives of the naturally sharp volume fraction field. With a higher degree of grid refinement, the volume fraction field becomes progressively sharper, which can lead to a lack of numerical convergence for curvature predictions. Within the two-phase flow community, a common procedure for addressing this error is the use of a level set function, $\phi(\mathbf{x}, t)$, which is much smoother than α and can subsequently provide more accurate curvature predictions. This combination of ϕ with α has been in existence for quite some time [12] and has also been adopted within OpenFOAM, for instance in the work of Ferrari et al. [7], Bilger et al. [9], Albadawi et al. [13], Menon et al. [14], Kassar et al. [15], Elsayed et al. [16], Vachaparambil and Einarsrud [17]. More explicitly in Ferrari et al. [7], Albadawi et al. [13], Menon et al. [14], the level set is initially constructed from the α field, namely by an equation of the form $\phi_o = (2\alpha - 1)\zeta$, where ζ is related to the local grid spacing, Δx. Subsequently, this initial ϕ_o is reinitialized through the solution of a Hamilton–Jacobi equation, resulting in a signed distance function, which is then used to compute curvature. In other work [9,17], the α field is smoothed out before calculating curvature. Other alternatives include the construction of a uniform voxel mesh followed by isosurface triangulation [16] or the representation of the interface by a cloud of points [15], both of which are used to generate more accurate predictions of κ.

The present work aims to examine the result of improvements to curvature estimation within `interFoam`. In contrast to previously published works, the evaluation exercises considered are more relevant to sprays and hydrodynamic breakup dynamics than bubbles or similar configurations. Particularly, the addition of the Rayleigh breakup problem and a 2-phase shear instability solved analytically through the Orr–Sommerfeld temporal analysis is aimed at these types of breakup scenarios.

Our approach is similar to the works mentioned previously in constructing a ϕ field from the α field and computing curvature from ϕ. However, there is a distinguishing characteristic in the present work in that ϕ was created from a dual treatment that treats interfacial cells (cells containing the $\alpha = 0.5$ isosurface) differently from the remaining cells. The motivation for this dual treatment lies in the erroneous displacement of the interface that results if the Hamilton-Jacobi reinitialization solution is applied in a domain that includes interfacial cells (cells that contain a segment of the interface). This problem has been documented previously [18,19] and it results from the propagation of information from the wrong side of the interface for some interfacial nodes. For the sake of providing an additional comparison, the common smoothing of the α field [9,17] is also implemented.

The contents are organized as follows. In Section 2, a description is provided for the three curvature prediction methodologies. The evaluation exercises are presented in Section 3 in order of increasing complexity. Since we are interested in liquid spray and droplet applications, the choice of exercises reflects this interest. A summary of our findings along with concluding remarks are presented in Section 4.

2. Curvature Predictions

The curvature implementations and flow predictions reported in the present work employ the algebraic solver `interFoam` (version 2.1.1). More recent versions of OpenFOAM under both ESI and foundation releases utilize essentially the same conventional curvature calculation found in OpenFOAM 2.1.1. A description of the liquid fraction transport, momentum, and pressure equation treatment can be found in [11]. Since the focus of the paper is on interfacial curvature, the primary relevant parameter is the surface tension force appearing in the momentum equation. Its implementation is performed via the Continuum Surface Force (CSF) method [20], namely

$$\int_{\Omega}\int_{\Gamma} \sigma\kappa(\mathbf{x}_s)\delta(\mathbf{x} - \mathbf{x}_s)\mathbf{n}_{\Gamma}\, d\Gamma(\mathbf{x}_s)d\mathbf{x} = \int_{\Gamma\cap\Omega} \sigma\kappa(\mathbf{x}_s)\mathbf{n}_{\Gamma} d\Gamma(\mathbf{x}_s) = \int_{\Omega} \sigma\kappa\nabla\alpha d\mathbf{x}. \quad (1)$$

Here, the terms on the left-hand side (LHS) and center correspond to the analytical expression of the surface tension force exerted over an arbitrary space $\Omega \subseteq \Re^d$ $(d = 2, 3)$. The interface is denoted by Γ (tacitly assumed to be temporally evolving), the curvature is denoted by $\kappa(\mathbf{x}_s)$, the surface tension coefficient is denoted by σ (treated as constant), the unit interface normal is denoted by $\mathbf{n}_\Gamma(\mathbf{x}_s)$ (oriented from the *gas to the liquid* dictated by the direction of $\boldsymbol{\nabla}\alpha$), and the 3D Dirac Delta function is denoted by $\delta(\mathbf{x} - \mathbf{x}_s)$. Its numerical treatment via the CSF is given by the term on the right-hand side (RHS) in Equation (1), where the local curvature is defined as

$$\kappa(\mathbf{x}_s) \equiv -\boldsymbol{\nabla} \cdot [\mathbf{n}_\Gamma(\mathbf{x}_s)]. \tag{2}$$

Since the present development is made within the framework of `OpenFoam`, its existing finite volume discretization is utilized. In particular, for an arbitrary scalar field $\gamma(\mathbf{x}, t)$ and vector field $\mathbf{v}(\mathbf{x}, t)$, the following gradient and divergence approximations are employed:

$$(\boldsymbol{\nabla}\gamma)_i = \frac{1}{|\Omega_i|} \sum_{f \in \partial\Omega_i} \gamma_{f,i} \mathbf{S}_{f,i}, \tag{3}$$

$$(\boldsymbol{\nabla} \cdot \mathbf{v})_i = \frac{1}{|\Omega_i|} \sum_{f \in \partial\Omega_i} (\mathbf{v}_{i,f} \cdot \mathbf{S}_{f,i}). \tag{4}$$

In these equations, Ω_i corresponds to the space occupied by some computational cell i and $|\Omega_i|$ is its respective volume. The corresponding surface area of cell i is given by $\partial\Omega_i$. Quantities pertaining to *cell centers are denoted with subscript i*, and *quantities pertaining to face quantities* are denoted with subscript f. The product of the surface area of the cell face f (associated with cell i) and the corresponding *outward* unit vector is given by $\mathbf{S}_{f,i}$. Equations (3) and (4) come directly from the Gauss Divergence Theorem and its gradient analog along with the application of the mean value theorem for definite integrals.

2.1. Standard Calculation (α-Based)

The standard method or α-based method is the native method present in `interFoam`. Even though the current version being employed is 2.1.1, other versions, including more recent ones such as ESI Openfoam-v2006 and OpenFoam Foundation version 6, continue to employ the same formulation. In the α-based method, the curvature is computed as follows:

1. Cell-centered liquid fractions are linearly interpolated to the faces, i.e., $\alpha_i \rightarrow \alpha_{f,i}$.
2. The computation of $(\boldsymbol{\nabla}\alpha)_i$ is performed using Equation (3), with the scalar function $\alpha(\mathbf{x}, t)$.
3. Normal vectors are calculated via

$$(\mathbf{n}_{\Gamma,\alpha})_i = \frac{(\nabla\alpha)_i}{|(\nabla\alpha)_i| + \delta}, \tag{5}$$

   where $\delta = 10^{-8}$ is a small scalar and its inclusion is to avoid division by zero. Calculation of $(\nabla\alpha)_i$ employs Equation (3).
4. Cell-centered values for $(\mathbf{n}_{\Gamma,\alpha})_i$ are linearly interpolated to cell faces, i.e., $(\mathbf{n}_{\Gamma,\alpha})_i \rightarrow (\mathbf{n}_{\Gamma,\alpha})_{i,f}$
5. Having face values for the interface normal, the curvature is calculated using Equation (4), namely

$$\kappa_i = -\frac{1}{\Omega_i} \sum_{f \in \Omega_i} (\mathbf{n}_{\Gamma,\alpha})_{i,f} \cdot \mathbf{S}_{f,i}. \tag{6}$$

The aforementioned framework is part of `OpenFoam` and works for general polyhedra, i.e., it is not limited to structured grids.

2.2. Diffused α-Based Calculation ($\tilde{\alpha}$-Based)

It is understood [21] that due to the sharp nature of α, taking second-order derivatives to compute curvature leads to gross errors. One obvious alternative for reducing the magnitude of the error is to smooth out the α field. As noted in the Introduction, some authors [9,17] smooth the field by filtering α or the curvature within OpenFOAM. In the review article by Popinet [1], he cites these methods as obsolete since newer methodologies offer better results. In this work, this smoothing methodology is used principally to provide another comparison data point against the standard method and the ϕ-based method discussed below. However, rather than explicitly smoothing α or filtering the curvature, in the present work, the smoothing is applied by solving an unsteady diffusion equation for the liquid fraction over a specified number of time steps. The process is as follows:

1. Initialize $\tilde{\alpha}$ with the α field at some arbitrary time level n, i.e.,

$$\tilde{\alpha}(\mathbf{x}_i, \tau = 0) = \alpha(\mathbf{x}_i, t_n), \tag{7}$$

where τ is pseudo time.

2. Advance $\tilde{\alpha}$ through the solution of

$$\frac{\partial \tilde{\alpha}}{\partial \tau} = \mathcal{D}\nabla^2\tilde{\alpha}, \tag{8}$$

where $\mathcal{D}$ is the diffusion coefficient. Since the aim is to smooth the initial liquid fraction field over a length scale $\ell = \Delta x$, the associated total computation time is $\tau_f = \Delta x^2/\mathcal{D}$. With such a short computation time, the choice is to employ an explicit methodology. The stability criterion in 2D is $\Delta x^2/(4\mathcal{D})$ and in 3D is $\Delta x^2/(6\mathcal{D})$; hence, at most, $O(10)$ time steps are required. Numerically, the solution is advanced via

$$\tilde{\alpha}_i(\tau_{n+1}) = \tilde{\alpha}_i(\tau_n) + \frac{\Delta\tau}{\Omega_i}\sum_{f\in\Omega_i}(\boldsymbol{\nabla}\tilde{\alpha}_{i,f}(\tau_n)\cdot\mathbf{S}_{f,i}). \tag{9}$$

3. Normal vectors at the end of the previous calculation are computed via

$$(\mathbf{n}_{\Gamma,\tilde{\alpha}})_i = \frac{(\boldsymbol{\nabla}\tilde{\alpha})_i}{|(\boldsymbol{\nabla}\tilde{\alpha})_i| + \delta}, \tag{10}$$

4. Cell-centered values for $(\mathbf{n}_{\Gamma,\tilde{\alpha}})_i$ are linearly interpolated to cell faces, i.e., $(\mathbf{n}_{\Gamma,\tilde{\alpha}})_i \rightarrow (\mathbf{n}_{\Gamma,\tilde{\alpha}})_{i,f}$
5. Curvature is calculated based on Equation (4) as

$$(\kappa_{\tilde{\alpha}})_i = -\frac{1}{\Omega_i}\sum_{f\in\Omega_i}(\mathbf{n}_{\Gamma,\tilde{\alpha}})_{i,f}\cdot\mathbf{S}_{f,i} \tag{11}$$

2.3. Distance-Based Calculation (ϕ-Based)

Motivated by the tradition of coupling level set with VoF to improve curvature estimate, in the present work, we similarly create this signed distance function, ϕ. As previously discussed in the Introduction, the solution of the reinitialization equation at interfacial cells causes the interface to be erroneously displaced within these cells. Hence, the present approach is implemented in two steps, namely (i) interface reconstructions at interfacial cells [22] and (ii) the solution of Hamilton–Jacobi equation excluding interfacial cells.

2.3.1. Interface Reconstruction

1. For cells with $\epsilon < \alpha_i < (1-\epsilon)$, the α_i values are linearly interpolated to cell vertices as illustrated in Figure 1, where $\epsilon = 0.01$.

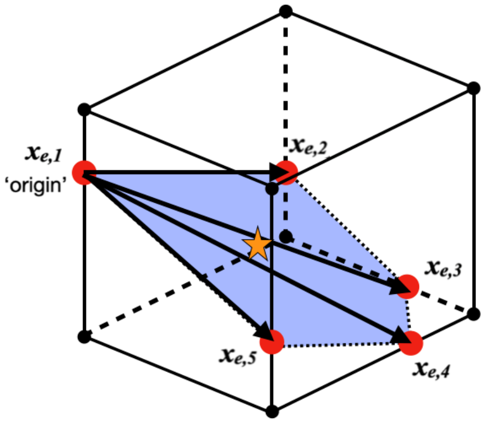


Figure 1. Illustration of the interface reconstruction step in the ϕ-based curvature calculation. Cell vertices are shown as black spheres and the polygon centroid is indicated by a star. Red spheres denote the location that corresponds to the interception of a cell edge with the local $\alpha = 0.5$ isosurface. These locations are denoted as interface edge points.

2. Using the cell-vertex values of α, the location along each cell edge that corresponds to $\alpha = 0.5$ is identified. These locations are denoted as interface edge points and are labeled as $\mathbf{x}_{e,j}$, as shown in Figure 1. The subscript e indicates that they correspond to cell edges, and the subscript j is the index of the points. If a cell has three or more interface edge points, the cell is identified as an *interfacial cell*. The polygon intersecting these points represents the local interface reconstruction within the interfacial cell.
3. The polygon area vector $\mathbf{A}_{\Gamma,i}$ is computed for each interfacial cell:
   (a) At each interfacial cell, one edge points is chosen as the origin. For instance, in Figure 1, $\mathbf{x}_{e,1}$ is the origin. The remaining points $\mathbf{x}_{e,k}$ ($k = 2, N_e$) are sorted in cyclical order, where N_e is the total number of interface edge point [22].
   (b) Computation of the area vector of the polygon by adding the area vectors for the triangles composing the polygon [23],

$$\mathbf{A}_{\Gamma,i} = \frac{1}{2}\sum_{j=1}^{N_e-1}\left((\mathbf{x}_{e,j} - \mathbf{x}_{e,1}) \times (\mathbf{x}_{e,j+1} - \mathbf{x}_{e,1})\right). \tag{12}$$

   To ensure that the area vector points from the liquid to the gas in all cells,

$$\text{if } \nabla\alpha \cdot \mathbf{A}_{\Gamma,i} > 0, \quad \text{then } \mathbf{A}_{\Gamma,i} \text{ is set to } -\mathbf{A}_{\Gamma,i}. \tag{13}$$

4. The polygon centroid, $\mathbf{x}_{\Gamma,c}$, is calculated from

$$\mathbf{x}_{\Gamma,c} = \frac{1}{N_e}\sum_{j=1}^{N_e}\mathbf{x}_{e,j}. \tag{14}$$

5. The perpendicular distance, $d_\perp$, from the polygon centroid to the center of the interfacial cell is computed as

$$d_\perp = \mathbf{r} \cdot \frac{\mathbf{A}_{\Gamma,i}}{|\mathbf{A}_{\Gamma,i}|}, \tag{15}$$

where $\mathbf{r} = \mathbf{x}_{\Gamma,c} - \mathbf{x}_{\text{cell cent}}$ and $\mathbf{x}_{\text{cell cent}}$ is the location of the cell center. For instance, $d_\perp$ is less than zero for $\mathbf{x}_{\text{cell cent.}}$ in the gas phase. Hence, the signed distance function ϕ is less than zero in the gas and greater than zero in the liquid. Thus, $\nabla\phi/|\nabla\phi|$ is into the liquid consistent direction of $\mathbf{n}_{\Gamma,\alpha}$ and $\mathbf{n}_{\Gamma,\tilde{\alpha}}$.

2.3.2. Hamilton–Jacobi Reinitialization

The solution of the reinitialization problem via the Hamilton–Jacobi equation is a well-established method for reinitializing a level set function to a distance function [24]. It consists of solving

$$\frac{\partial\phi}{\partial\tau} = S(\phi_0(\mathbf{x}))(1 - |\nabla\phi|), \tag{16}$$

where $\phi_0(\mathbf{x}) = \phi(\mathbf{x}, \tau = 0)$ and $S(\ldots)$ denotes a sign-function, i.e., $S(x) = 1$ for $x > 0$, $S(x) = 0$ for $x = 0$, and $S(x) = -1$ for $x < 0$. An equivalent form of Equation (16), which offers some insights into the behavior of this equation is

$$\frac{\partial\phi}{\partial\tau} + S(\phi_0)\frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla\phi = S(\phi_0). \tag{17}$$

In this Lagrangian form, the propagation speed of ϕ is clearly shown as $S(\phi_0)\frac{\nabla\phi}{|\nabla\phi|}$, and its evolution occurs in pseudo time. The process implemented in the current work is as follows:

1. Initialization of ϕ obtained from

$$\phi(\mathbf{x}_i, \tau = 0) = \begin{cases} d_\perp, & \text{for interfacial cells} \\ [\alpha(\mathbf{x}_i, t_n) - 0.5]\Delta x, & \text{for the rest of the cells.} \end{cases} \tag{18}$$

This results in $\phi(\mathbf{x}_i, \tau = 0) = \Delta x/2$ in liquid cells and $\phi(\mathbf{x}_i, \tau = 0) = -\Delta x/2$ in gas cells.

2. Explicit solution of the Hamilton–Jacobi equation (Equation (16)) via finite volume

$$\frac{\phi(\mathbf{x}_i, \tau + \Delta\tau) - \phi(\mathbf{x}_i, \tau)}{\Delta\tau} = \left[S(\phi_0(\mathbf{x}_i)) - \frac{S(\phi_0(\mathbf{x}_i))}{|\Omega_i|}\int_{\Omega_i}|\nabla\phi_i^n|dV\right]\mathbb{I}_{int}(\mathbf{x}_i), \tag{19}$$

where

$$\mathbb{I}_{int}(\mathbf{x}_i) = \begin{cases} 0, & \text{for interfacial cells} \\ 1, & \text{for the rest of the cells.} \end{cases} \tag{20}$$

Since $\nabla\phi_i^n$ has only one degree of freedom inside a given cell,

$$\int_{\Omega_i}|\nabla\phi_i^n|dV = \left|\int_{\Omega_i}\nabla\phi_i^n dV\right|. \tag{21}$$

Approximating the integral above, we have

$$\int_{\Omega_i}\nabla\phi_i^n dV = \sum_f(\phi_f^n\mathbf{S}_f) = \sum_f[\phi_{f,up}^n + \mathbf{r}\cdot\nabla\phi_{up}^n]\mathbf{S}_f, \tag{22}$$

where $\phi_{f,up}^n$ is the upwind value of ϕ^n and $\mathbf{r} = \mathbf{x}_i - \mathbf{x}_{i,f}$. To obtain these upwind values, the propagation velocity and surface normal vectors are used, i.e., $S(\phi_0)\frac{\nabla\phi}{|\nabla\phi|}\cdot S_f$ for each f belonging to cell i. Furthermore, $\nabla\phi_{up}^n$ is approximated by a central differencing scheme evaluated at the cell center, where the values of $\phi_{f,up}^n$ are employed at cell faces. Putting things together, we then have

$$\phi(\mathbf{x}_i,\tau+\Delta\tau)=\phi(\mathbf{x}_i,\tau)+S(\phi_0(\mathbf{x}_i))\Delta\tau\left[1-\frac{1}{|\Omega_i|}\left|\sum_f[\phi^n_{f,up}+\mathbf{r}\cdot\boldsymbol{\nabla}\phi^n_{up}]\mathbf{S}_f\right|\right]\mathbb{I}_{int}(\mathbf{x}_i). \tag{23}$$

Thus, interfacial cells are, by construction, left unchanged at their initial value of $\phi(\mathbf{x}_i)=d_\perp$. The pseudo-time step in the solution of Equation (23), $\Delta\tau$, is proportional to the grid resolution, and $\Delta\tau=0.2\Delta x$ is presently used. The solution of Equation (23) is extended to *a few cells around the interface* to provide more than sufficient support for the numerical stencil employed in the calculation of κ, i.e., there is no need for the calculation to extend throughout the domain. In the computations presented here, eight cells on either side of the interface are employed.

3. Computation of normal vector via

$$\mathbf{n}_{\phi,i}=\frac{\nabla\phi(\mathbf{x}_i)}{|\nabla\phi(\mathbf{x}_i)|+\delta} \tag{24}$$

is performed over all cells and again $\delta=10^{-8}$. The computation of $\boldsymbol{\nabla}\phi(\mathbf{x}_i)$ employs Equation (3).

4. Computation of κ. The cell-centered normal vectors are interpolated to cell faces $\mathbf{n}_{\phi,i}\rightarrow(\mathbf{n}_{\phi,i})_f$, and using Equation (4)), the curvature is calculated as

$$(\kappa_\phi)_i=-\frac{1}{\Omega_i}\sum_{f\in\Omega_i}(\mathbf{n}_{\phi,i})_f\cdot\mathbf{S}_{f,i}. \tag{25}$$

3. Results

The performance of the three curvature estimation methods is examined among five evaluation exercises as shown in Table 1. Since the main focus of the work concerns curvature improvement, which is directly related to surface tension, the choice of problems is, for the most part, heavily influenced by this effect. The only exception is the temporal interfacial instability analysis at a Weber number of 100 presented in Section 3.3.

Table 1. Evaluation exercises considered in the present work.

Section 3.1	Laplace Pressure Problem
Section 3.2	3D Oscillating Droplet
Section 3.3	Shear Layer: Growth of Interfacial Instability
Section 3.4	Rayleigh–Plateau Instability
Section 3.5	Retraction of Liquid Column

3.1. Laplace Pressure Problem

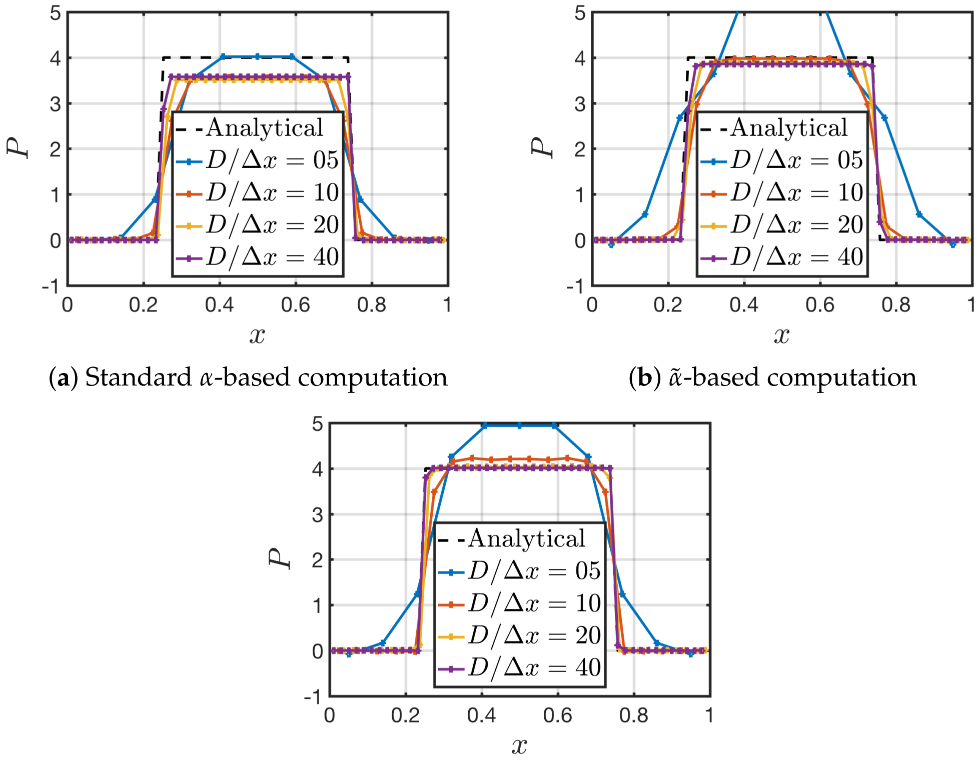
The Laplace pressure test is undertaken to isolate the accuracy of surface tension predictions. This case is governed by the Young–Laplace equation, namely $\Delta P=P_L-P_G=\sigma\kappa$, where the pressure on the liquid and gas side are, respectively, given by P_L and P_G. Here and in the remainder of the document, the liquid and gas phases are denoted by subscripts, L, and G, respectively.

In the present case, the exercise consists of a liquid 2D droplet in zero gravity having a radius $R=0.25$ m, and $\kappa=4$ m^{-1}, densities $\rho_L=\rho_G=10^4$ kg/m^3, viscosity $\mu_L=\mu_G=1$ kg/(ms), and surface tension $\sigma=1$ kg/s^2 yielding a Laplace number

$$La=\frac{\sigma\rho_L R}{\mu_L}=2500. \tag{26}$$

The domain consists of a $[4R\times4R]$ square with the droplet positioned at its geometric center. The radius of the droplet is $R=0.25$. For boundary conditions, a zero gradient is imposed for α, a zero Dirichlet condition for pressure, and partial derivatives of all velocity components with respect to the coordinate normal for each boundary are zero. This is termed a zero gradient condition for velocity. Due to surface tension, a jump in pressure is immediately recorded, which initiates a transient. Since the calculation is performed well within the numerically stable regime and physically corresponds to a steady solution, the transient decays and an equilibrium condition is achieved. The examination of pressure jump is performed during this steady period, and consequently, the computations are executed to $t = 125$ s consistent with previous studies in the literature [11,25]

The pressure profile along the centerline is extracted and plotted in Figure 2 corresponding to the three methods for computing κ. In each plot, the analytical solution is provided along with predictions corresponding to four different levels of numerical resolution. For the standard α-based method shown in Figure 2a, it can be seen that, upon refinement, the pressure inside the liquid converges to a value lower than the analytical value, in agreement with an earlier publication [11]. This is a direct result of the poor curvature predictions, as expected. The pressure values improve for the other two methods, as seen in Figure 2b,c. For the $\tilde{\alpha}$ curvature method shown in Figure 2b, the results are noticeably better, leading to what appears to be an agreement with the exact values. For the ϕ-based computation, the results show that predictions with a resolution finer than $D/\Delta x = 20$ are essentially indistinguishable from analytical values.



(**a**) Standard α-based computation

(**b**) $\tilde{\alpha}$-based computation

(**c**) ϕ-based computation

Figure 2. Pressure distribution along the centerline of a 2D droplet ($R = 0.25$) is presented at four different grid resolutions corresponding to the three curvature schemes.

To quantify the degree of convergence, the Laplace pressure is computed for all three methods as a function of grid resolution $D/\Delta x$. This pressure is computed according to

$$\Delta P = P_{in} - P_{out}, \tag{27}$$

where P_{in} and P_{out} are, respectively, the internal and external pressure to the droplet and are computed as follows

$$P_{out} = \frac{1}{N_{V_{out}}} \sum_{\Omega_{i,j} \in V_{out.}} P(\mathbf{x}_{i,j}), \tag{28}$$

$$P_{in} = \frac{1}{N_{V_{in}}} \sum_{\Omega_{i,j} \in V_{in}} P(\mathbf{x}_{i,j}), \tag{29}$$

where V_{out} is the region given by $r > 3D/4$ and V_{in} is the region given by $r < D/4$. The number of computational cells in each region is given by $N_{V_{out}}$ and $N_{V_{in}}$. The reason for this choice of regions is to ensure that, with coarse level calculations, the interior and exterior domains are properly accounted for. The associated numerical error is

$$\epsilon_{\Delta P} = |\Delta P - \Delta P_{exact}|. \tag{30}$$

This error is calculated as a function of grid resolution in Figure 3. Two reference lines corresponding to first- and second-order convergences are included as well. Beginning with the standard method, the predictions initially diverge and reach a plateau at an appreciable high level of error. No convergence is observed. For the $\tilde{\alpha}$, the magnitude of the error is smaller; however, the method also lacks convergence. Lastly, for the ϕ-based method, second-order convergence is achieved, and the method is superior to the previous two schemes for computing curvature.

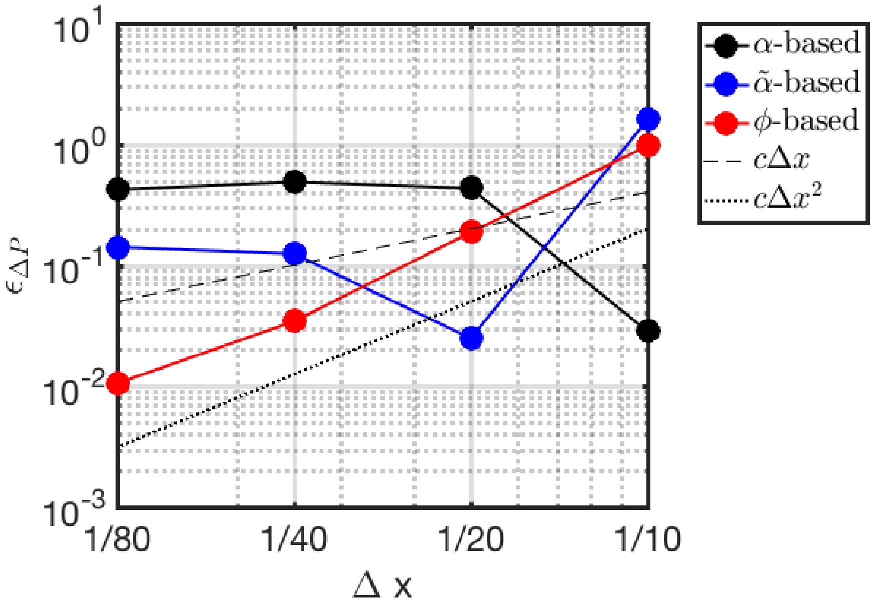


Figure 3. Laplace pressure error for the 2D droplet corresponding to the three curvature schemes. The ϕ-based method is the only one that produces convergent results to approximately the second order.

3.2. 3D Oscillating Droplet

This exercise consists of introducing a small perturbation on an otherwise quiescent spherical droplet under zero gravity, where the resulting motion is driven by surface tension. Thus, the present exercise aims to isolate the effects of curvature predictions in a dynamic environment. If the unperturbed radius is given by R_0, the initial surface of the droplet is given by

$$R(\theta, t=0) = R_0[1 + \eta P_2(\cos\theta)] = R_0\left(1 + \frac{\eta}{2}\left[3(\cos\theta)^2 - 1\right]\right), \tag{31}$$

where θ is the polar angle given by $\theta = \cos^{-1}(z/\sqrt{x^2+y^2+z^2})$ and $R(\theta, t)$ is the distance of the interface from the center of the droplet at some (θ, t). An illustration of this problem is shown in Figure 4.

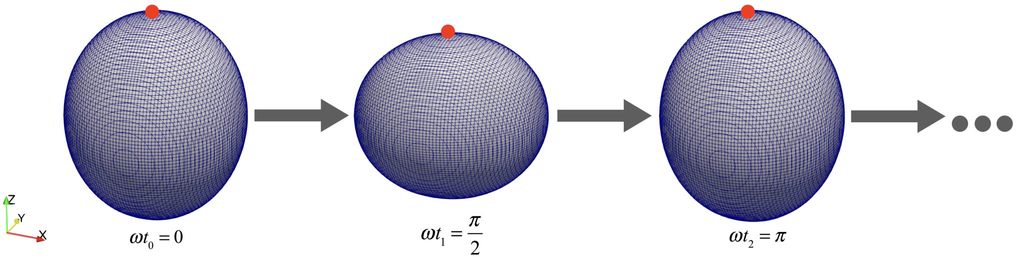


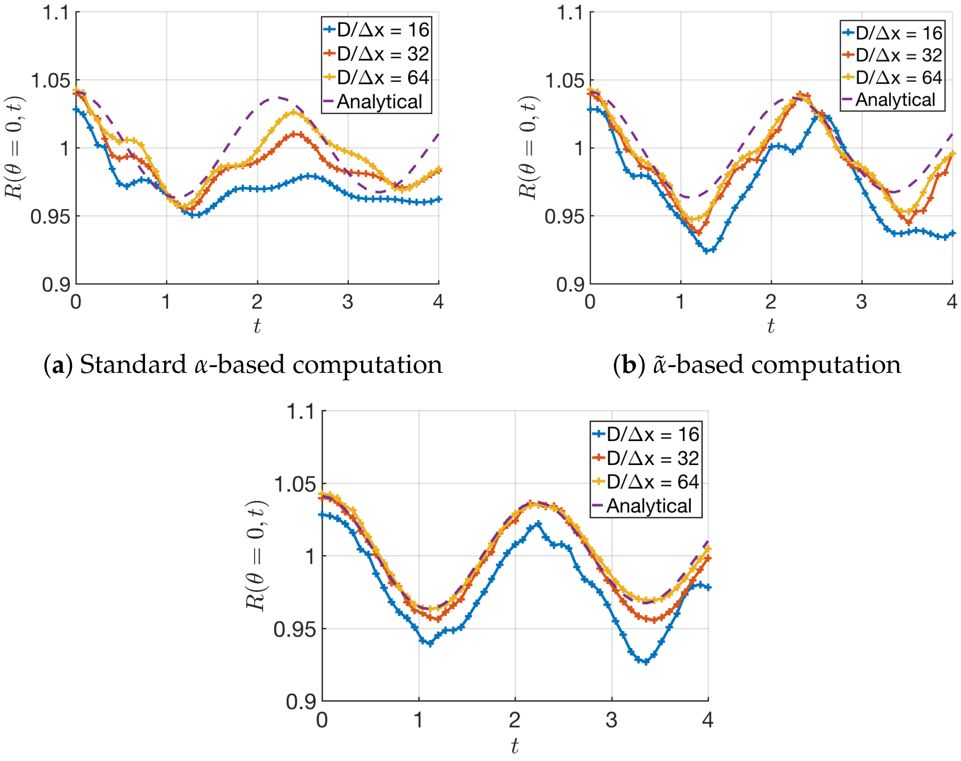
Figure 4. Illustration of the droplet oscillation problem.

In the present exercise, $R_0 = 1$ m and the magnitude of the perturbation is $\eta = 0.04$ m. The departure from pure spherical form is described by the Legendre polynomial of second order, $P_2(\cos\theta)$. The droplet is positioned at the geometric center of a $[4R_0 \times 4R_0 \times 4R_0]$ domain. For boundary conditions, a zero gradient is assigned for α and pressure, and a no-slip velocity condition is imposed at all the boundaries. The densities employed are $\rho_L = 1$ kg/m^3 and $\rho_G = 0.01$ kg/m^3; the viscosities are $\mu_L = 0.01$ kg/(m s) and $\mu_g = 0.0001$ kg/(m s), and the surface tension is $\sigma = 1$ kg/s^2. This setup is the same as the one used by Patel et al. [26] and the resulting Laplace number $(2R\rho_L\sigma/\mu_L^2)$, La =10,000, which is characterized by the dominant effect of the product of inertia and surface tension forces over the square of viscous forces.

In the work of Lamb [27] (pg. 563) , an analytical expression for the motion of the interface is provided. A more modern version of this expression can be found in Patel et al. [26], namely

$$R(\theta = 0, t) = R_0 + \eta \exp\left(-t\frac{5\mu_L}{\rho_L R_0^2}\right)\cos\left(t\sqrt{\frac{24\sigma}{R_0^3[3\rho_L + 2\rho_G]}}\right). \tag{32}$$

Predictions from the three curvature schemes are compared against this analytical solution, as shown in Figure 5. The first observation is that predictions at different levels of resolution do not agree initially, i.e., at $t = 0$. This is particularly the case for the coarser case, i.e., $D/\Delta x$ =16, and is due to the degree of numerical error in identifying the interface ($\alpha = 0.5$ contour). At higher levels of resolution, the agreement at $t = 0$ improves substantially. A second observation is that the solution from the standard α-based method contains erroneous harmonics that do not disappear even at the finest level of resolution. The agreement with the exact solution is far from encouraging. Employing the $\tilde{\alpha}$-based method, there is some measure of improvement, but still, the numerical predictions contain anomalous oscillations. In the ϕ-based method, anomalous behavior is slightly observed in the coarse case, but with higher levels of resolution, the predictions quickly match the analytical solution and no erroneous harmonics populate the predictions.



(**a**) Standard α-based computation

(**b**) $\tilde{\alpha}$-based computation

(**c**) ϕ-based computation

Figure 5. Predictions of the displacement of the interface at the top of the 3D droplet, $R(\theta = 0, t)$, employing the three methodologies at different levels of numerical resolution.

To quantify the level of performance, the following error metric is employed

$$\epsilon_R = \frac{1}{N_{t_i}} \sum_{t_i = t_0}^{t_i = t_{end}} \frac{|R(\theta = 0)_{num}(t_i) - R(\theta = 0)_{exact}(t_i)|}{R(\theta = 0)_{exact}(t_i)}, \quad (33)$$

where t_i is the discrete-time level and N_{t_i} is the total number of time steps. The numerical prediction for each method is denoted by $R(\theta = 0)_{num}(t_i)$. As such, this error represents a time-averaged of the instantaneous deviation from the analytical solution. The results are shown in Figure 6 as a function of grid resolution, and a first-order reference line ($c\Delta x$) is provided. The observation made previously is confirmed, where it is apparent that the ϕ-based method has a lower error magnitude compared with the other two methods. The convergence rate for all three methods is initially better than first-order accuracy but slows down at higher resolution levels. Interestingly, the standard α-based case shows convergence even though, visually, the result does not appear to be encouraging.

With the inclusion of dynamics, all three methods display numerical convergence, with the ϕ-based method having a consistent lowest magnitude of the error. However, the convergence rate for the ϕ-based method is no longer second order but has declined to approximately first order, most likely due to the associated errors involved in calculating momentum and displacement of the interface. It is interesting to see, however, that this added motion had a negative effect on the convergence rate of the most accurate method presented in this work but a positive effect in the less accurate methods. In fact, for this case, these methods have improved from having no convergence in the static case to convergence in the present dynamic exercise.

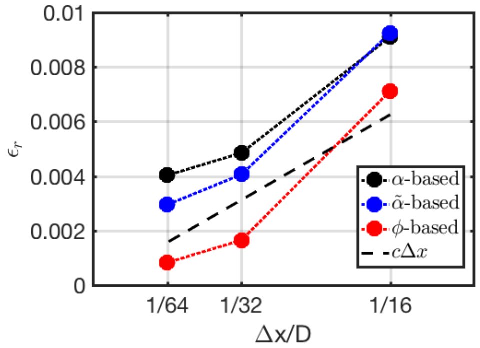


Figure 6. Numerical error corresponding to the three curvature estimation methods as a function of grid resolution for the problem of a 3D oscillating droplet.

3.3. Shear Layer: Growth of Interfacial Instability

The two-phase shear layer is a fundamental fluid mechanics problem that is particularly relevant to atomization and spray formation and is often omitted in numerical tests for curvature predictions. This problem is illustrated in Figure 7, and it consists of a gas and liquid stream that is characterized by a velocity difference leading to the establishment of a shear layer. Authoritative descriptions of the dynamics characterized by Kelvin–Helmholtz instability can be found in the text by Criminale et al. [28] (pg. 39–42).

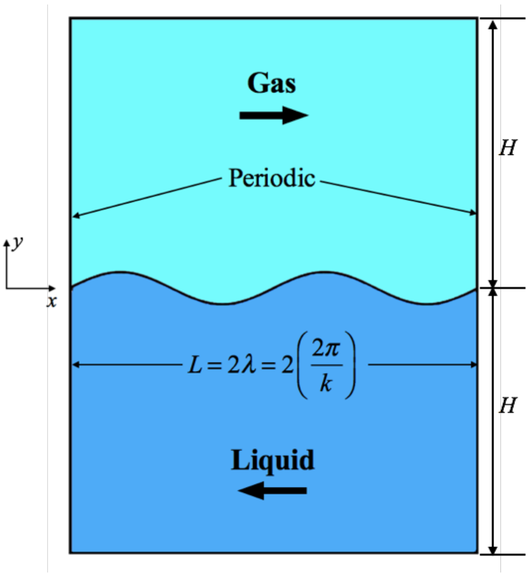


Figure 7. Illustration of the shear layer setup with base flow motion in the gas and liquid phase regions in opposite directions and with an imposed perturbation on the gas–liquid interface.

In the present work, a 2D temporal instability Orr–Sommerfeld calculation is employed, where a perturbation of wavenumber $k = 2\pi/\lambda$ (λ is the wavelength) is imposed with periodic boundary conditions on the left and right sides of the domain, as shown in Figure 7. On the top and bottom boundaries, a zero gradient is imposed for α and pressure and a slip condition is imposed for velocity. This slip velocity condition consists of zero gradient for velocity components, which are tangential to the bottom and top boundaries, i.e., x-direction, along with a zero normal velocity component.

In brief, the velocity field is decomposed as

$$\mathbf{u}(x,y,t) = [u(x,y,t), v(x,y,t)] = \mathbf{U}(y) + \mathbf{u}'(x,y,t), \tag{34}$$

where $\mathbf{U}(y) = [U(y), 0]$ is the base flow profile and $\mathbf{u}'(x, y, t) = [u'(x, y, t), v'(x, y, t)]$ is the perturbation velocity. The base profile used in this exercise is

$$U_L = U_L^* erf\left(\frac{y}{\delta_L}\right) \quad y < 0, \tag{35}$$

$$U_G = U_G^* erf\left(\frac{y}{\delta_G}\right) \quad y > 0, \tag{36}$$

with values of U_L^* =0.99 m/s and U_G^* =1 m/s. These values can be obtained from the shear stress balance of the base flow field at the interface, which gives $\mu_L U_L^*/\delta_L = \mu_G U_G^*/\delta_G$. The perturbation flow velocity is described in terms of a stream function, $\psi(x, y, t)$, as

$$\mathbf{u}'(x, y, t) = \nabla \times \psi \mathbf{e}_z, \tag{37}$$

where the stream function takes the normal mode form of

$$\psi(x, y, t) = \mathbb{R}\{\varphi(y) \exp(ik[x - ct])\}. \tag{38}$$

In this expression, $\mathbb{R}$ extracts the real part of its argument, the eigenvector $\varphi = \varphi_R + i\varphi_i \in \mathbb{C}$, the input wavenumber $k = k_R \in \mathbb{R}$ ($k_I = 0$), and the complex wave speed $c = c_R + ic_I \in \mathbb{C}$. The imaginary part of the wave speed, c_I, is responsible for the growth or decay of a given mode. The normal mode expression for the stream function is substituted into the Orr–Sommerfeld equation in combination with the gas–liquid interfacial constraints, namely continuity of normal and horizontal velocity and balance of tangential and normal stresses. Details are provided in [29]. The resulting linear system is a generalized eigenvalue problem that is solved for $((\varphi(y), c))$, where $\varphi(y)$ is the y-profile of the stream function and c is the complex eigenvalue.

The predictions from the Orr-Sommerfeld system are compared in this exercise to those obtained from the VoF employing the aforementioned methods for estimating curvature. The domain is 2D of length $L = 2\lambda$ and height $H = 6$ m. The fluid densities are given by $\rho_L = 1\ \mathrm{kg/m^3}$ and $\rho_G = 0.1\ \mathrm{kg/m^3}$, the viscosities are given by $\mu_L = 5.05 \times 10^{-5}\ \mathrm{kg/(m\,s)}$ and $\mu_G = 5 \times 10^{-5}\ \mathrm{kg/(m\,s)}$, the surface tension coefficients are given by $\sigma = 0.01\ \mathrm{kg/s^2}$ and $\sigma = 0.1\ \mathrm{kg/s^2}$, and $g = 0\ \mathrm{m/s^2}$. The associated Weber number for this exercise is defined as

$$We = \frac{\rho_L U_G^{*2} \delta_G}{\sigma}, \tag{39}$$

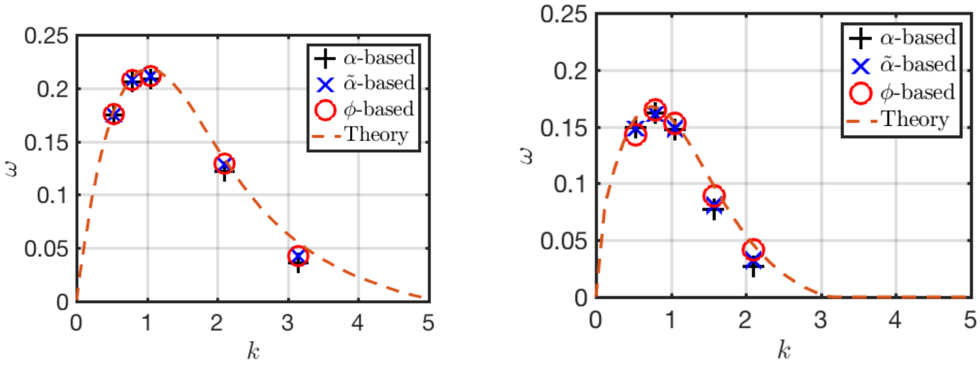
hence, the above two values for the surface tension coefficient correspond to two cases considered, i.e., $We = 100$ and $We = 10$.

In the calculations, the grid refinement is set to $\Delta x = 0.011$ m to ensure that the small initial perturbation is sufficiently resolved. A convenient way to track the growth of the perturbation is to compute its kinetic energy. For the type of periodic domains considered here, this method bypasses the need to compute the perturbation growth by meticulously trying to fit the interface at various times during the growth process. The perturbation kinetic energy, $KE_{pert}(t)$, is computed as

$$KE_{pert}(t) = \int_\Omega \frac{\rho}{2} [\mathbf{u}'(\mathbf{x}, t) \cdot \mathbf{u}'(\mathbf{x}, t)] dV, \tag{40}$$

where $\Omega = \Omega_L \cup \Omega_G$, $\Omega_L = [0, 2\lambda] \times [-H, 0^-]$ (liquid domain), $\Omega_G = [0, 2\lambda] \times [0^+, H]$ (gas domain) (see Figure 7), and $\rho = \rho_L + (1 - \alpha)\rho_G$. It can be shown [29] that $KE_{pert}(t) = (B_L + B_G) \exp(2\omega t)$, where B_L and B_G are constants. Thus, $KE_{pert}(t)/KE_{pert}(0) = \exp(2\omega t)$ and the value of the growth rate, ω, can be readily obtained from plots of this kinetic energy ratio. From Equation (38), the growth rate is related to c via $\omega = -kc_i$. The comparisons between VoF predictions and linear stability results are shown in Figure 8 and in Tables 2 and 3.

In the tables, the error is calculated as $100 \times \left|\omega_{num} - \omega_{analytical}\right| / \omega_{max,analytical}$, where the normalization factor is $\omega_{max,analytical}$, i.e., the maximum growth rate. This normalization factor is employed to ensure that the same factor applies to all errors calculated. For instance, a normalization factor equal to $\omega_{analytical}$ would result in an exaggerated reporting of the error at high wave numbers even for cases where the error is relatively minor due to the growth rate approaching zero in this part of the wave number domain.



(**a**) We = 100 (**b**) We = 10

Figure 8. Comparison of the growth rate of the shear-layer instability against analytical solutions.

Table 2. Percentage error ($100 \times \left|\omega_{num} - \omega_{analytical}\right| / \omega_{max,analytical}$) in perturbation growth rate predictions for $We = 100$.

	$k = \pi/6$	$k = \pi/4$	$k = \pi/3$	$k = 2\pi/3$	$k = \pi$
α-based	1.3	1.4	4.2	4.0	2.3
$\tilde{\alpha}$-based	1.2	1.0	3.0	2.4	1.1
ϕ-based	0.9	0.7	3.0	2.5	0.9

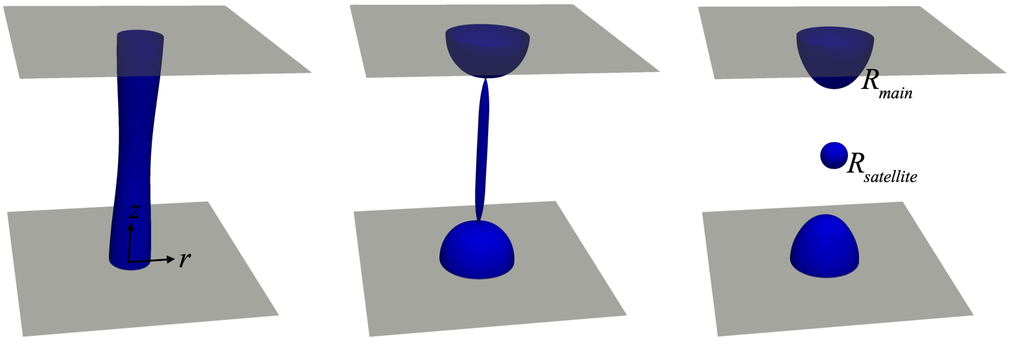
Table 3. Percentage error ($100 \times \left|\omega_{num} - \omega_{analytical}\right| / \omega_{max,analytical}$) in perturbation growth rate predictions for $We = 10$.

	$k = \pi/6$	$k = \pi/4$	$k = \pi/3$	$k = \pi/2$	$k = 2\pi/3$
α-based	4.2	3.0	4.0	12.4	11.8
$\tilde{\alpha}$-based	5.0	3.5	3.4	10.5	8.2
ϕ-based	8.2	1.0	0.5	5.7	3.2

Overall good agreement is observed between the simulation results and the theoretical prediction with the three methods. Due to the stabilizing action of surface tension, the cases for $We = 10$ exhibit a slower growth rate than those at $We = 100$. At higher We, all errors are within 5% but increase to a maximum of 12.4% at $We = 10$. This is not surprising since the influence of surface tension and, thus, the effect of curvature error increase with lower values of We. Additionally, the magnitude of the error decreases when transitioning from the standard α-based method to the ϕ-based method. However, this improvement is not as dramatic as in the previous exercises. The predictions are reasonably good even with the standard method, particularly at We =100.

3.4. Rayleigh–Plateau Instability

Another exercise that is relevant to hydrodynamic breakup consists of a Rayleigh–Plateau instability [30]. This is a surface tension-driven flow jet breakup under zero gravity, which is similar to ligament breakup during spray formation [31]. The initial stage is characterized by a liquid column with a varicose sinusoidal perturbation imposed on its surface, as shown in Figure 9a. This perturbation grows and eventually fragments the liquid column, as seen in Figure 9b,c. The breakup leads to the formation of two structures, a large main droplet or parent droplet at the periodic boundary of the domain and a smaller satellite droplet in the central part of the domain. The periodic nature of the simulation domain in the vertical direction implies that the main droplet is continuous across the top and the bottom faces of the domain.



(**a**) Initial setup (**b**) Growth of perturbation (**c**) After breakup

Figure 9. The evolution of the Rayleigh breakup of a liquid column leading to the final configuration consisting of a main droplet and a single satellite droplet. The top and bottom faces constitute periodic boundaries.

Lafrance [32] provides a detailed theoretical analysis and experimental data of the breakup of the laminar liquid jet accounting for non-linear effects. We compare the present results against these theoretical predictions. All simulations are performed in a 3D domain with dimensions $[\lambda \times \lambda \times \lambda]$, where λ is the wavelength of the disturbance imposed. Periodic conditions are applied on the top and bottom boundaries. For all the other boundaries, zero gradient condition is imposed for α and velocity, and zero Dirichlet for pressure. The simulations are based on a water jet of $R_0 = 17.5 \times 10^{-6}$ m with $\rho_L = 1000$ kg/m^3, $\mu_L = 10^{-3}$ kg/(m s), $\rho_G = 1$ kg/m^3, $\mu_G = 0$ kg/(m s), and σ =0.073 kg/s^2. The spatial resolution is given by $D/\Delta x = 25.5$ in the region surrounding the liquid jet. Away from the jet a coarser grid is employed. The initial perturbation, which is described by the radial distance of the interface from the axis of the jet, is given by

$$R(z, t=0) = R_0 + A\cos\left(k\frac{z}{R_0}\right), \tag{41}$$

where $A = 0.15R_0$ and k is the dimensionless perturbation wavenumber, given by $k = 2\pi R_0/\lambda$. Simulations with $k = 0.5, 0.6$ and 0.75 are performed.

The radii of the resulting droplets are presented in Figure 10 along with the theoretical predictions [32]. The top and bottom curves in this plot represent, respectively, the main droplet size and satellite droplet size. The theoretical predictions are shown using dotted lines, and the simulation results are shown with different markers. Quantification of the error is provided by

$$\epsilon_r(k) = \frac{\left|r(k) - r_{theory}(k)\right|}{r_{theory}(k)} \times 100, \tag{42}$$

where $r(k)$ is the numerically predicted main or satellite radii and $r_{theory}(k)$ is the theoretical counterpart. This error data are presented in Tables 4 and 5.

The main droplet size predictions corresponding to the three methods are close enough to each other that the markers in Figure 10 practically overlap. For the satellite droplet predictions, the differences are more noticeable. This is expected since the volume of the main droplet is so much larger than the satellite droplet that numerical errors affect satellite size predictions more strongly. Among the three methods, the ϕ-based computations lead to the best predictions consistent with previous tests; however, the results for all three methods are not drastically different. Additionally, the errors have about the same magnitude at the different wavenumbers examined. It was observed that, for some cases, the $\tilde{\alpha}$-based method is prone to a slight departure from asymmetry, which leads to the formation of more than one satellite droplet. This raises some questions regarding the suitability of this method in breakup problems.

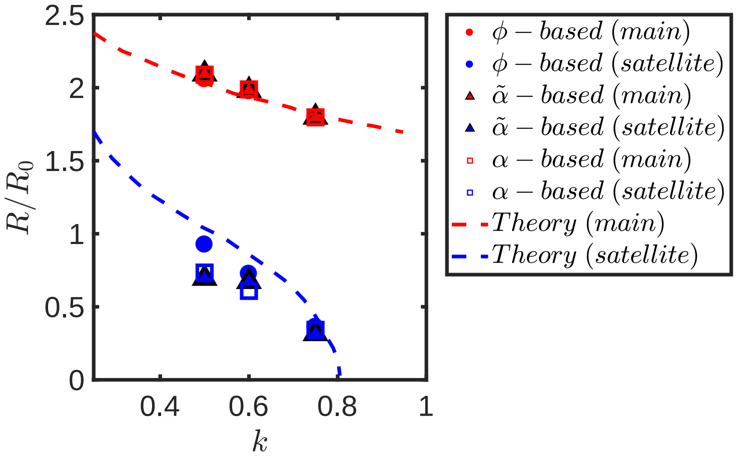


Figure 10. Comparison between main and satellite droplet radii between simulations and theoretical predictions from [32]. Theoretical predictions are in dashed lines, and the top and bottom curves are, respectively, the main and satellite droplet calculations.

Table 4. Percentage error, $\epsilon_r(k)$, in main droplet size predictions.

	k = 0.5	k = 0.6	k = 0.75
α-based	2.6	2.8	0.9
$\tilde{\alpha}$-based	3.1	2.4	0.9
ϕ-based	1.3	2.1	1.0

Table 5. Percentage error, $\epsilon_r(k)$, in satellite droplet size predictions.

	k = 0.5	k = 0.6	k = 0.75
α-based	29.2	29.0	21.5
$\tilde{\alpha}$-based	33.0	21.6	27.5
ϕ-based	10.8	15.4	17.0

3.5. Retraction of Liquid Column

The last exercise consists of a surface tension drive retraction of a liquid column under zero gravity conditions presented by Umemura [33]. In this last test, there are no analytical solutions and, thus, the evaluation is restricted to a comparison among the three prediction methods. The conditions employ a liquid (SF$_6$) with a density of $\rho_L = 1460$ kg/m^3, a viscosity of $\mu_L = 1.1 \times 10^{-3}$ kg/(m s), and a surface tension of 1.605×10^{-3} kg/s^2. The gas is pressurized nitrogen with a density and viscosity given, respectively, by $\rho_G = 79.1$ kg/m^3 and $\mu_G = 1.76 \times 10^{-5}$ kg/(m s). The domain has dimensions [0.8424 mm × 0.8424 mm × 5.1168 mm] with the axis of the liquid column aligned with the z-axis and positioned at the center of the left face. This left face is a wall, with zero gradient conditions for pressure and α, and a no-slip velocity condition. At all other boundaries, a zero gradient is applied for α and velocity, and a zero Dirichlet condition is applied for pressure. The cylindrical column has an initial radius $R = 0.12748$ mm and length 5.1168 mm.

The simulations are performed on a uniform mesh with a resolution of $\Delta x = 7.8$ µm yielding a jet diameter to grid size ratio of $D/\Delta x$ =32. At this level of resolution, the results for column length time history are numerically converged. Its time evolution, as shown in Figure 11, is non-dimensionalized by the capillary time scale, which is given by $\sqrt{(\rho_L R^3)/\sigma}$ [34] (pg. 3;)thus, $t^* = t\sqrt{\frac{\sigma}{\rho_L R^3}}$, where R is the radius of the cylinder. In this non-dimensionalization, the wave number $k = 1/R$ has been used along with the fact that $\rho_G \ll \rho_L$, and thus, ρ_G can be ignored.

The perfectly cylindrical liquid column at $t^* = 0$ starts retracting, and a bulb and a neck are formed at the tip, as shown on the figures at $t^* = 7.8$. With the persistent action of surface tension, the bulb continues to grow, as depicted at $t^* = 18.9$. At this middle time, subtle differences appear between prediction from the standard method and the other two methods. At the final time of t^* = 30, the predictions between the ϕ-based and $\tilde{\alpha}$-based are essentially the same and are slightly different from the standard method.

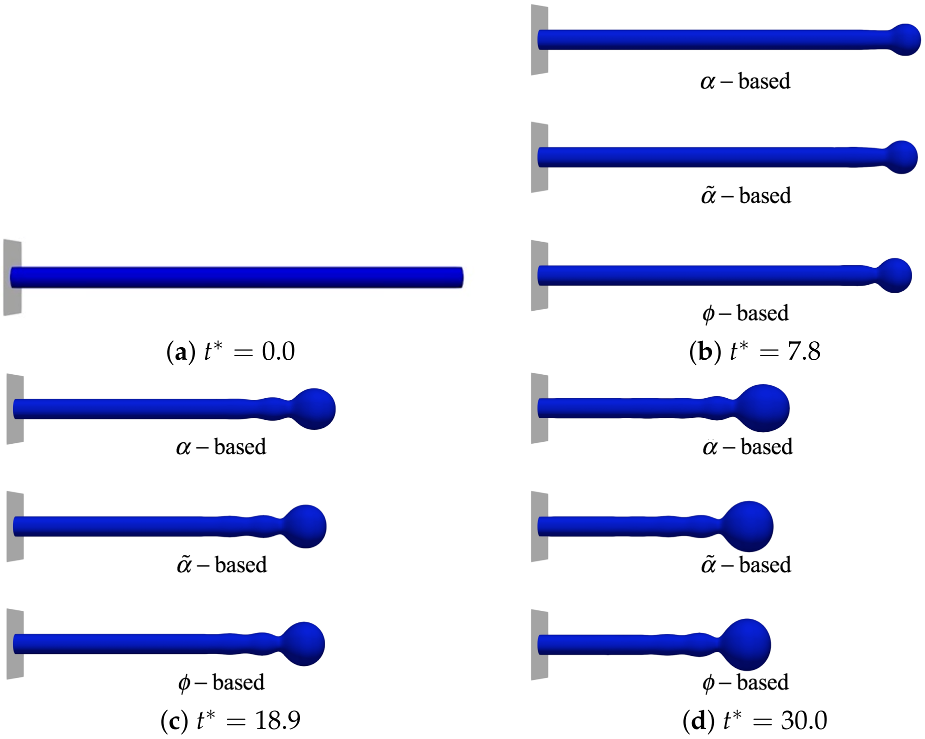


Figure 11. The evolution of the retracting liquid column corresponding to the three curvature estimation schemes.

The quantitative difference among the three curvature estimation methods is provided by comparing the time history of the liquid column lengths, as shown in Figure 12. Initially, the length of the liquid column remains unaltered as only the periphery is affected by the dynamics, but the central part remains largely unchanged. The flat predictions of column length reflect this during the initial period, which lasts until approximately $t^* = 3.2$. Beyond this initial period, the retraction of the liquid column is essentially a linear function of time, implying that the retraction speed is constant. The lengths from the diffused-α based method and the ϕ-based method are almost identical and are slightly faster than the α-based computation. While subtle differences emerge in the shapes of the bulbs and the lengths of the liquid columns predicted by the three methods, the differences in the overall dynamics appear minor. An improved curvature scheme does not impact the results in a significant way.

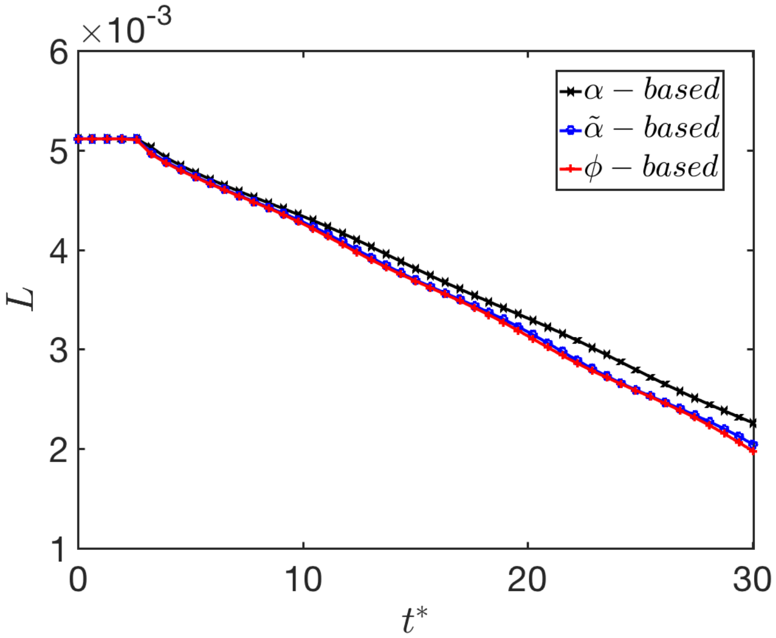


Figure 12. The time evolution of the column length predicted by the different curvature estimation schemes.

3.6. Evaluation of Computational Costs

The calculations in the previous sections have shown that the ϕ-based method is noticeably superior to the standard α-based calculation, particularly when the effect of surface tension is isolated from all other factors. This section documents the computational cost of exercising this ϕ-based method, where our findings are presented in Table 6 as a percentage of the total cost for all operations. Specifically, the ϕ-based calculation consists of an interface reconstruction and a Hamilton–Jacobi re-initialization, as outlined in Section 2.3. In contrast, the standard α-based method has no such interface reconstruction or re-initialization costs.

Table 6. Distribution of computational costs as a percentage of total costs. Quantities presented are average values corresponding to 3500 time steps.

	α Equation	**Velocity Equation**	**Pressure Equation**	**ϕ-Based Calculation**	**Overhead**
Laplace pressure problem	29.8%	0.89%	54.31%	11.67%	3.33%
3D oscillating drop	9.62%	4.21%	57.53%	27.5%	1.14%
Shear layer problem	2.61%	0.94%	89.86%	6.45%	0.14%
Rayleigh-Plateau instability	10.42%	5.53%	46.28%	32.73%	5.04%
Retraction of liquid column	9.71%	5.86%	46.93%	36.71%	0.79%

For the sake of completeness, the cost of all other operations are also included in Table 6. This consists of the α transport operation, which consists of the solution of the VoF equation. It also includes the momentum equation, which is the predictor part of the calculation, and the Pressure–Poisson equation, which usually dominates all computational costs. Overhead costs are also included. The results show that the Pressure Poisson calculation remains, not surprisingly, the dominant contributor to the computational expense. However, the ϕ-based operations are not necessarily negligible and, in some instances, are relatively close to the Pressure calculation. This means that, in the worst-case scenarios, the calculations involving the current ϕ-based method can be up to 50% more expensive than the standard α-based methodology. However, it should be kept in mind that no effort has been made in expediting these additional ϕ-based calculations at this stage. It remains part of future work.

4. Conclusions

Improvements to the interfacial curvature calculations have been implemented and tested against the standard α-based methodology employed in `interFoam`. The new implementations consist of a diffused treatment for the liquid fraction ($\tilde{\alpha}$-based) and a signed-distance function procedure (ϕ-based). Five evaluation exercises have been performed, emphasizing hydrodynamic breakup problems, and are thus relevant to spray formation and other atomization processes.

In summary, our findings show that, among the three methods, the ϕ-based method is consistently superior in all cases, with the most notable result being the second-order convergence of the Laplace pressure across a 2D droplet. As evident in the literature, the lack of convergence for the standard α-based method has been known for quite some time, and the $\tilde{\alpha}$ shows a lower level of error but still fails to converge. An interesting finding is that the predictions converge even for the standard method once the problem involves motion, such as the oscillating droplet case. Additionally, the predictions for the shear layer (Section 3.3), the Rayleigh jet breakup (Section 3.4), and the retraction of a liquid column (Section 3.5) show that the standard method does not differ significantly from the more accurate ϕ-based method, even though its performance is worse. The difference between α-based and ϕ-based methods is accentuated when surface tension effects become the dominant influencing factor, for instance, for the 3D oscillating droplet.

An interesting conclusion from the present work is that, once other factors enter into the dynamics, such as viscous forces, inertia, or pressure, the performance of the standard α-based treatment becomes quite reasonable. For instance, the agreement with theory in the droplet oscillating case (Section 3.2), the shear layer (Section 3.3), and the Rayleigh jet (Section 3.4) are for the most part acceptable. It is only when the evaluation exercise focuses entirely on the curvature prediction, for instance, the Laplace pressure problem (Section 3.1) or the spurious current estimations presented in the literature [1,26,35] that we see the standard method falling quite short of the exact result. However, most practical cases concerning liquid breakup do not involve such carefully orchestrated scenarios, and in these more general cases, the standard method does a reasonable job. A takeaway from the current work is that adding more complicated and accurate curvature prediction methods to `interFoam` does not necessarily guarantee *meaningful* improvements in realistic flow simulations unless those simulations are to a great degree governed by surface tension, e.g., nano-droplet dynamics.

Author Contributions: A.A. conducted the research and investigation process, specifically developed and implemented code, executed simulations, and performed validation. He was also responsible for providing visualization and data presentation. M.A. executed simulations, provided additional help in creating figures, and helped with the preparation of the paper. M.F.T. is responsible for the conceptualization of the project, its administration, funding acquisition, supervision, and writing of the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Direct-Injection Engine Research Consortium (DERC) at the University of Wisconsin-Madison.

Acknowledgments: We are grateful for the support provided by the Direct-Injection Engine Research Consortium (DERC) at UW-Madison. Additionally, the authors acknowledge the computing access provided through the Extreme Science and Engineering Discovery Environment (XSEDE) Bridges regular memory at the Pittsburgh Super-computing Center under allocation TG-CTS180037.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Popinet, S. Numerical models of surface tension. *Annu. Rev. Fluid Mech.* **2018**, *50*, 49–75. [CrossRef]
2. Ubbink, O.; Issa, R. A method for capturing sharp fluid interfaces on arbitrary meshes. *J. Comput. Phys.* **1999**, *153*, 26–50. [CrossRef]
3. Rusche, H. Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions. Ph.D. Thesis, Imperial College London (University of London), London, UK, 2003.

4. Roenby, J.; Bredmose, H.; Jasak, H. A computational method for sharp interface advection. *R. Soc. Open Sci.* **2016**, *3*, 160405. [CrossRef] [PubMed]
5. Gamet, L.; Scala, M.; Roenby, J.; Scheufler, H.; Pierson, J.L. Validation of volume-of-fluid OpenFOAM® isoAdvector solvers using single bubble benchmarks. *Comput. Fluids* **2020**, *213*, 104722. [CrossRef]
6. Omar, S. Development of a Hyperbolic Equation Solver and the Improvement of the OpenFOAMR Two-Phase Incompressible Flow Solver. Ph.D. Thesis, Cardiff University, Cardiff, UK, 2018.
7. Ferrari, A.; Magnini, M.; Thome, J.R. A Flexible Coupled Level Set and Volume of Fluid (flexCLV) method to simulate microscale two-phase flow in non-uniform and unstructured meshes. *Int. J. Multiph. Flow* **2017**, *91*, 276–295. [CrossRef]
8. Dianat, M.; Skarysz, M.; Garmory, A. A Coupled Level Set and Volume of Fluid method for automotive exterior water management applications. *Int. J. Multiph. Flow* **2017**, *91*, 19–38. [CrossRef]
9. Bilger, C.; Aboukhedr, M.; Vogiatzaki, K.; Cant, R. Evaluation of two-phase flow solvers using Level Set and Volume of Fluid methods. *J. Comput. Phys.* **2017**, *345*, 665–686. [CrossRef]
10. Popinet, S. Gerris: A tree-based adaptive solver for the incompressible Euler equations in complex geometries. *J. Comput. Phys.* **2003**, *190*, 572–600. [CrossRef]
11. Deshpande, S.S.; Anumolu, L.; Trujillo, M.F. Evaluating the performance of the two-phase flow solver interFoam. *Comput. Sci. Discov.* **2012**, *5*, 014016. [CrossRef]
12. Sussman, M.; Puckett, E. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *J. Comput. Phys.* **2000**, *162*, 301–337. [CrossRef]
13. Albadawi, A.; Donoghue, D.; Robinson, A.; Murray, D.; Delauré, Y. Influence of surface tension implementation in volume of fluid and coupled volume of fluid with level set methods for bubble growth and detachment. *Int. J. Multiph. Flow* **2013**, *53*, 11–28. [CrossRef]
14. Menon, S.; Nagawkar, J.; Nilsson, H. *Coupled Level-Set with VOF InterFoam*; Report; Chalmers University: Goteborg, Sweden, 2016.
15. Kassar, B.B.; Carneiro, J.N.; Nieckele, A.O. Curvature computation in volume-of-fluid method based on point-cloud sampling. *Comput. Phys. Commun.* **2018**, *222*, 189–208. [CrossRef]
16. Elsayed, O.; Kirsch, R.; Osterroth, S.; Antonyuk, S. An improved scheme for the interface reconstruction and curvature approximation for flow simulations of two immiscible fluids. *Int. J. Multiph. Flow* **2021**, *144*, 103805. [CrossRef]
17. Vachaparambil, K.J.; Einarsrud, K.E. On sharp surface force model: Effect of sharpening coefficient. *Exp. Comput. Multiph. Flow* **2021**, *3*, 226–232. [CrossRef]
18. Russo, G.; Smereka, P. A Remark on Computing Distance Functions. *J. Comput. Phys.* **2000**, *163*, 51–67. [CrossRef]
19. Anumolu, L.; Trujillo, M.F. Gradient augmented reinitialization scheme for the level set method. *Int. J. Numer. Methods Fluids* **2013**, *73*, 1011–1041. [CrossRef]
20. Brackbill, J.; Kothe, D.B.; Zemach, C. A continuum method for modeling surface tension. *J. Comput. Phys.* **1992**, *100*, 335–354. [CrossRef]
21. Williams, M.; Kothe, D.; Puckett, E. Chapter Accuracy and convergence of continuum surface tension models. In *Fluid Dynamics at Interfaces*; Cambridge University Press: Cambridge, UK, 1998; pp. 294–305.
22. Ryddner, D.T. Advection of a Reconstructed aVoF Method with Phase Change. Ph.D. Thesis, University of Wisconsin-Madison, Madison, WI, USA, 2018.
23. Braden, B. The Surveyor's Area Formula. *Coll. Math. J.* **1986**, *17*, 326–337. [CrossRef]
24. Osher, S.; Fedkiw, R. *Level Set Methods and Dynamic Implicit Surfaces*; Springer: Berlin/Heidelberg, Germany, 2003.
25. Hysing, S. A new implicit surface tension implementation for interfacial flows. *Int. J. Numer. Methods Fluids* **2006**, *51*, 659–672. [CrossRef]
26. Patel, H.; Kuipers, J.; Peters, E. Computing interface curvature from volume fractions: A hybrid approach. *Comput. Fluids* **2018**, *161*, 74–88. [CrossRef]
27. Lamb, H. *Hydrodynamics*; University Press: London, UK, 1924.
28. Criminale, W.; Jackson, T.; Joslin, R. *Theory and Computation in Hydrodynamic Stability*; Cambridge University Press: Cambridge, UK, 2003.
29. Deshpande, S.S.; Gurjar, S.R.; Trujillo, M.F. A computational study of an atomizing liquid sheet. *Phys. Fluids* **2015**, *27*, 082108. [CrossRef]
30. Rayleigh, L. On the instability of jets. *Proc. Lond. Math. Soc.* **1878**, *1*, 4–13. [CrossRef]
31. Deshpande, S.; Trujillo, M.; Kim, S. Computational study of fluid property effects on the capillary breakup of a ligament. *At. Sprays* **2013**, *23*, 1167–1195. [CrossRef]
32. Lafrance, P. Nonlinear breakup of a laminar liquid jet. *Phys. Fluids* **1975**, *18*, 428–432. [CrossRef]
33. Umemura, A. Self-destabilizing mechanism of a laminar inviscid liquid jet issuing from a circular nozzle. *Phys. Rev. E* **2011**, *83*, 046307. [CrossRef] [PubMed]
34. Hoeve, W.; Gekle, S.; Snoeijer, J.; Versluis, M.; Brenner, M.; Lohse, D. Breakup of Diminutive Rayleigh Jets. *Phys. Rev. E* **2010**, *22*, 122003.
35. Vachaparambil, K.J.; Einarsrud, K.E. Comparison of Surface Tension Models for the Volume of Fluid Method. *Processes* **2019**, *7*, 542. [CrossRef]