

PAPER

Evaluating the performance of the two-phase flow solver interFoam

To cite this article: Suraj S Deshpande *et al* 2013 *Comput. Sci. Discov.* **5** 014016

View the [article online](#) for updates and enhancements.

You may also like

- [A multi-scale geometric flow method for molecular structure reconstruction](#)
Guoliang Xu, Ming Li and Chong Chen
- [Direct numerical simulations of a thin liquid film coating an axially oscillating cylindrical surface](#)
Matthias Binz, Wilko Rohlfis and Reinhold Kneer
- [On the influence of inlet perturbations on slug dynamics in horizontal multiphase flow—a computational study](#)
S Schmelter, S Knotek, M Olbrich et al.

Evaluating the performance of the two-phase flow solver interFoam

Suraj S Deshpande, Lakshman Anumolu and Mario F Trujillo

Department of Mechanical Engineering, University of Wisconsin, Madison, WI 53706, USA

E-mail: mtrujillo@wisc.edu

Received 2 May 2012, in final form 5 October 2012

Published 9 November 2012

Computational Science & Discovery **5** (2012) 014016 (36pp)

[doi:10.1088/1749-4699/5/1/014016](https://doi.org/10.1088/1749-4699/5/1/014016)

Abstract. The performance of the open source multiphase flow solver, *interFoam*, is evaluated in this work. The solver is based on a modified volume of fluid (VoF) approach, which incorporates an interfacial compression flux term to mitigate the effects of numerical smearing of the interface. It forms a part of the C++ libraries and utilities of OpenFOAM and is gaining popularity in the multiphase flow research community. However, to the best of our knowledge, the evaluation of this solver is confined to the validation tests of specific interest to the users of the code and the extent of its applicability to a wide range of multiphase flow situations remains to be explored. In this work, we have performed a thorough investigation of the solver performance using a variety of verification and validation test cases, which include (i) verification tests for pure advection (kinematics), (ii) dynamics in the high Weber number limit and (iii) dynamics of surface tension-dominated flows. With respect to (i), the kinematics tests show that the performance of *interFoam* is generally comparable with the recent algebraic VoF algorithms; however, it is noticeably worse than the geometric reconstruction schemes. For (ii), the simulations of inertia-dominated flows with large density ratios $\sim \mathcal{O}(10^3)$ yielded excellent agreement with analytical and experimental results. In regime (iii), where surface tension is important, consistency of pressure–surface tension formulation and accuracy of curvature are important, as established by Francois *et al* (2006 *J. Comput. Phys.* **213** 141–73). Several verification tests were performed along these lines and the main findings are: (a) the algorithm of *interFoam* ensures a consistent formulation of pressure and surface tension; (b) the curvatures computed by the solver converge to a value slightly (10%) different from the analytical value and a scope for improvement exists in this respect. To reduce the disruptive effects of spurious currents, we followed the analysis of Galusinski and Vigneaux (2008 *J. Comput. Phys.* **227** 6140–64) and arrived at the following criterion for stable capillary simulations for *interFoam*: $\Delta t \leq \max(10\tau_\mu, 0.1\tau_\rho)$ where $\tau_\mu = \mu\Delta x/\sigma$, and $\tau_\rho = \sqrt{\rho\Delta x^3/\sigma}$. Finally, some capillary flows relevant to atomization were simulated, resulting in good agreement with the results from the literature.

Contents

1. Introduction	2
2. Numerical method	5
2.1. Evolution of liquid fractions	5
2.2. Solution of the momentum equation	8
3. Performance tests on interFoam	11
3.1. Kinematics tests	11
3.2. Dynamic tests for inertia-dominated flows	17
3.3. Tests for surface tension-dominated flows	19
3.4. Simulation of capillary flows	27
4. Conclusion	32
Acknowledgments	33
References	33

1. Introduction

Using highly resolved simulations to study multiphase flow, where the gas–liquid interface is sharply resolved, has become a relatively recent alternative for analyzing in great detail the physics of such flows. This is particularly the case in situations where hydrodynamic breakup or atomization is a dominant feature, for instance in the near field of an atomizing diesel jet [2] or in the air entrainment process produced by a plunging jet [3]. Attempts to successfully simulate various multiphase flow situations have resulted in a variety of numerical approaches [4]; among these, implicit interface capturing methods, namely volume of fluid (VoF) [5] and level set [6], have attracted significant attention as viable options for simulating flows involving extensive topology changes.

Among the numerical approaches and codes that are used to simulate multiphase flows, an open source VoF methodology named *interFoam* has been receiving an increased amount of attention and usage, as was documented recently in the literature [3, 7–17]. The first implementation of this VoF solver was done by Ubbink [18] in the framework of FOAM [19]. Since its inception, it has undergone several modifications. The present code forms part of a suite of C++ libraries of OpenFOAM, which is designed for finite-volume discretization of a generic set of partial differential equations generally, but not necessarily, linked to computational mechanics problems. The object-oriented techniques of C++ allow the code to closely resemble its mathematical counterpart and makes the top-level syntax amenable to development and modification [20]. In addition, several useful features such as ease of parallelization, availability of pre- and postprocessing utilities, error checks, choice of spatial and temporal discretization schemes, etc make the code attractive for research. Examples of some of the latest reported exercises of *interFoam* along with the type of code evaluation are described in table 1.

Despite the wide use of *interFoam*, its evaluation has for the most part been confined to validation exercises specific to the particular application of interest to the respective authors, as summarized in table 1. This calls to question the performance of the code for the spectrum of multiphase flow problems. In response to this, the present paper documents a detailed analysis of the behavior of *interFoam* over a wide range of verification and validation exercises.

An inspection of the literature concerning interface capturing methodologies (see, e.g., [1, 4, 5, 26–49]) reveals that at least the following four issues are fundamental to the accuracy of interfacial capturing methodologies.

- (i) *Accurate interface representation and advection.* In the framework of implicit interface capturing methods, the interface location, its normal and curvature are known only implicitly, from the underlying color function. For instance, in the VoF method, the interface is characterized by a discontinuous change in the volume fractions and difficulties related to its advection are well documented in the literature [27]. The VoF methods are known to exhibit a propensity to cause excessive diffusion of the interface, which is

Table 1. Recent multiphase flow studies using *interFoam*.

Authors and publication year	Evaluation of the solver
Berberovic <i>et al</i> [7] (2009)	Compared the process of droplet impact and crater formation with their experiments
Maiwald and Schwarze [8] (2011)	Simulated the plane plunging jets and compared the results for critical conditions for air entrainment with the experiments of [21]
Saito <i>et al</i> [9] (2011)	Simulated the modulated jets and compared the atomization characteristics with the experiments of [22]
Liu and Garcia [10] (2008)	Compared the velocity profiles for a flow around a partially submerged cylinder with the experimental data
Ishimoto <i>et al</i> [11] (2010)	With a modified version of <i>interFoam</i> , simulated the cavitation-assisted atomization in the gasoline injection system. Pictorially compared the computed instantaneous spray profiles and breakup behavior with their experiments
Raach <i>et al</i> [12] (2011)	Implemented an energy equation in the framework of <i>interFoam</i> . The solver was used without surface tension modeling to simulate heat transfer in a film falling over turbulence wires. Validation consisted of comparing phase velocities and wave peak heights with their experiments [23]
Roisman <i>et al</i> [13] (2010)	Simulated the droplet impact on a porous surface and qualitatively compared droplet spreading with their experiments
Saha and Mitra [14] (2009)	Compared microfluidic capillary flows with their experiments, with respect to dynamic and static contact angles
Saha <i>et al</i> [15] (2009)	Presented a comparison of a microfluidic flow in rectangular channels with an analytical solution of [24]
Gopala <i>et al</i> [16] (2011)	Implemented a model for Bingham plastics. Verified their model against analytical solutions for velocity profile in channel flow. Validated the solver with a flowability test in a V-funnel
Trujillo <i>et al</i> [17] (2011)	Implemented an energy equation in <i>interFoam</i> for computing droplet impact heat transfer. Validated crown formation with the experiments of [25] and temperatures with their own experiments
Deshpande <i>et al</i> [3] (2012)	Compared the velocity and liquid fraction profiles in a plunging jet flow with their experiments

highly undesirable. As has been precisely pointed out in [27], such a diffusion is a result of naive advection of the underlying, discontinuous VoF function. Attempts to alleviate this issue have resulted in two classes of advection methodologies—geometric and algebraic. The geometric reconstruction techniques were introduced in the late 1970s and have undergone several improvements since then [30–33]. The basic idea there is to explicitly reconstruct a sub-cell representation of the interface using an assumed functional form and then advect it in a Lagrangian sense to accurately compute volume fluxes. While these methods definitely prevent numerical diffusion of the interface, there still exist some unresolved issues with them, such as the difficulty of implementation in higher dimensions and increased computational expense [46], which have prompted continuous investigation. The second class of methods completely eliminates the reconstruction procedure and instead uses algebraic manipulations such as high-resolution or compressive differencing schemes [47, 50]. The recent improved Tangent of Hyperbola Interface Capturing with Slope

Weighting (THINC-SW) scheme of [46] also belongs to this category. While considerable simplicity of algorithm and computational saving result from avoiding reconstruction, the resulting interface is spread over two to three cells rather than being precisely known.

Apart from accurate advection of the VoF function, difficulties arise in the estimation of interface curvature due to the discontinuous nature of the volume fraction field, and in this regard, several alternatives, including smoothing of the volume fraction field and construction of height function, have been suggested [35, 51]. While these methods appear to perform well in mildly behaved test cases, several authors [32, 52] have indicated that these methods also become error-prone in high-curvature or poorly resolved regions.

- (ii) *Mass conservation (VoF and level set methods)*. The VoF methods, by construction, conserve mass. However, as discussed above, maintaining a sharp and accurate interface with VoF methods is far from trivial. Level set methods offer a more accurate representation of the interfacial quantities such as interfacial normal and curvature. This is desirable especially in two-phase flows involving surface tension effects. First used in the context of multiphase incompressible flows by Sussman *et al* [26], the level set methods have matured into a promising numerical technique for accurate simulation of multiphase flows [4, 53]. However, a well-known shortcoming of this approach has been that mass conservation is not imbedded in the formulation [26]. As a result, high-curvature or poorly resolved regions of the flow are typically susceptible to such ‘mass loss’. Significant improvements have been made in this regard through the development of techniques such as the use of marker particles interface advection [48, 54], coupling level sets and VoF methods [49, 55] and a gradient augmented level set method [56]. Another pending issue is the loss of signed distance property due to advection errors. To restore this property, an additional re-initialization procedure is required [26, 57]. This not only increases the computational cost, but can also bring about an artificial displacement of the interface resulting in mass loss/gain [58–61].
- (iii) *Spurious currents*. Almost all the implicit interface capturing methods are known to be susceptible to generation of unphysical flow at the interface solely due to numerical inadequacies. The main sources of spurious currents have been identified as inaccurate interface curvature and lack of a discrete force balance [1]. While the first one is related to the choice of interface tracking methodology, the latter relates to a consistent formulation of the flow solver. In this regard, the following strategies have been employed to mitigate this shortcoming: (a) improvements to curvature estimation [1, 37, 62], (b) ensuring a discrete balance between pressure gradient and surface tension [1, 63, 64], (c) sharp representation of the interface using the ghost fluid approach in conjunction with accurate curvature estimation [49, 65], (d) time resolution of capillary flow generated at the stair-stepped interface [28, 66] in an explicit implementation of the surface tension and (e) temporally semi-implicit treatment of surface tension [34]. While a generation of spurious velocities is not a major issue for inertia-dominated flows, it poses serious concerns in the computation of capillary flows, which generally characterize the small liquid structures in hydrodynamic flows involving atomization.
- (iv) *Handling large density ratios*. Simulating flows with large density ratios is known to be a difficult task [27]. Inconsistencies in the formulation of continuity and momentum equations are seen to result in unphysical flows including artificial deformation of the interface [67]. As a solution to this problem in the VoF framework, Rudman [62] proposed the use of consistent volume fluxes in both mass and momentum advection equations. Raessi and Pitsch [43] extended this idea to level set methods, using geometric arguments. Recently, Ghods and Herrman [44] proposed a density corrected scheme for their level set method, also based on the theme of consistent mass–momentum formulation.

In what follows, the algorithm of *interFoam* is first described, paying attention to the handling of VoF advection and momentum solution. This is followed by a series of tests on the solver, which cover the issues discussed above. Section 3.1 contains the verification tests on the advection algorithm of *interFoam*. The solver performance is then evaluated for the dynamics of inertia-dominated flows (section 3.2) and surface tension-dominated flows (section 3.3). The issue of spurious currents and their control in *interFoam* is discussed in sections 3.3.3 and 3.3.4, respectively. Finally, computational results relevant to atomization are presented, followed by a summary of our findings.

2. Numerical method

In this section we describe the solution procedure employed in `interFoam`, a two-phase incompressible flow solver in OpenFOAM. All the descriptions in this section are based on our analysis of the code itself and the relevant library files. We note that although we have tested version 1.5 in this work, the current results extend to the recent versions up to OpenFOAM-2.1.1, as the methodology, which is based on a compressive interfacial treatment of the liquid fraction advection, remains unchanged at its core. Independent of its implementation in `interFoam`, this technique merits thorough validation and verification due to its robustness and speed in handling problems with significant degree of topological changes. The solver employs finite volume discretization on collocated grids for the solution of two-phase equations of flow. The flow variables are cell centered, but their face interpolated values are also used in the solution procedure.

2.1. Evolution of liquid fractions

Considering a physical domain, Ω , composed of a liquid-phase and a gas-phase region, respectively, denoted as \mathfrak{R}_l and \mathfrak{R}_g , we can define an indicator as

$$\mathbb{I}(\mathbf{x}, t) = \begin{cases} 1 & \text{for } \mathbf{x} \in \mathfrak{R}_l \text{ at time } t, \\ 0 & \text{for } \mathbf{x} \in \mathfrak{R}_g \text{ at time } t. \end{cases} \quad (1)$$

Due to the singular nature associated with such a function, one method that presents an attractive discretization route is the integration of this quantity over the volume corresponding to a computational cell (Ω_i). This yields a liquid fraction field given by

$$\gamma(\mathbf{x}_i, t) = \frac{1}{|\Omega_i|} \int_{\Omega_i} \mathbb{I}(\mathbf{x}, t) dV. \quad (2)$$

Hence, for cells that are completely liquid or completely gas, the liquid fraction $\gamma(\mathbf{x}_i, t)$ is, respectively, equal to one or zero.

From conservation of mass, we have the following governing equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\mathbf{U} \rho) = 0, \quad (3)$$

where the indicator function above can be used to describe the density field as

$$\rho(\mathbf{x}, t) = \rho_l \mathbb{I}(\mathbf{x}, t) + \rho_g (1 - \mathbb{I}(\mathbf{x}, t)). \quad (4)$$

Here ρ_l and ρ_g adopt constant values in each respective phase. Introducing this expression into equation (3), we obtain

$$(\rho_l - \rho_g) \frac{\partial \mathbb{I}(\mathbf{x}, t)}{\partial t} + \nabla \cdot ((\rho_l - \rho_g) \mathbb{I}(\mathbf{x}, t) \mathbf{U}(\mathbf{x}, t)) + \rho_g \nabla \cdot \mathbf{U}(\mathbf{x}, t) = 0. \quad (5)$$

Integrating the above equation over Ω_i for a divergence free flow field results in

$$\int_{\Omega_i} (\rho_l - \rho_g) \frac{\partial \mathbb{I}(\mathbf{x}, t)}{\partial t} dV + \int_{\Omega_i} \nabla \cdot ((\rho_l - \rho_g) \mathbb{I}(\mathbf{x}, t) \mathbf{U}(\mathbf{x}, t)) dV = 0. \quad (6)$$

Following the definition of liquid volume fraction from equation (2) and using the Leibniz integral rule, equation (6) can be rewritten as

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot (\mathbf{U} \gamma) = 0. \quad (7)$$

This is the equation that is essentially solved in `interFoam`, albeit with some needed modifications to avoid the serious numerical diffusion issues. We should note that the velocity field appearing in this expression originates from a liquid filtering operation stemming from the second term on the left-hand side (lhs) of equation (5). Physically, it implies that the transport of the liquid fraction field has to be accomplished by a velocity field that corresponds to the liquid. In the gas-phase region, γ is zero, and respectively the liquid filtering of the velocity is likewise zero.

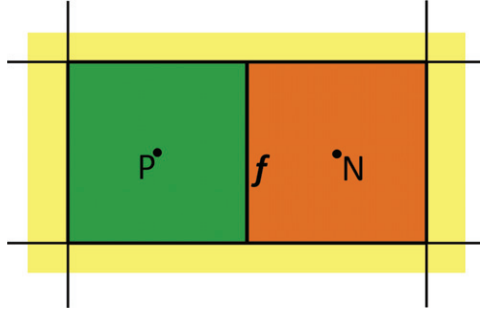


Figure 1. The face f whose owner is P and neighbor N .

By the discontinuous nature of the indicator function and to a great degree the liquid fraction field through equation (2), the transport of γ must preserve this sharpness. This is an important constraint that must be met by any numerical scheme designed to solve equation (7). In a common version of the VoF method [27], this is accomplished by sub-grid level reconstruction of the gas–liquid interface using linear or quadratic polynomials [32, 68, 69]. In *interFoam*, the approach takes a different route by modifying the advection term itself. We describe this approach by first rewriting equation (7) in the integral form as

$$\int_{\Omega_i} \frac{\partial \gamma}{\partial t} dV + \int_{\partial \Omega_i} \gamma \mathbf{U} \cdot \mathbf{n} dS = 0. \quad (8)$$

Its discrete counterpart is given by

$$\frac{\gamma_i^{n+1} - \gamma_i^n}{\Delta t} = -\frac{1}{|\Omega_i|} \sum_{f \in \partial \Omega_i} (F_u + \lambda_M F_c)^n, \quad (9)$$

where the unsteady term is discretized using the forward Euler scheme, and the advection term appears as a summation over the cell faces (f) of Ω_i . Other discrete approximations to the unsteady term include implicit Euler, Crank-Nicholson and backward differencing schemes. We will consider only the forward Euler in the present description. Quantities evaluated at the faces are subscripted by f , as shown in figure 1. The advective fluxes F_u and F_c are expressed by

$$F_u = \phi_f \gamma_{f,\text{upwind}} \quad \text{and} \quad (10)$$

$$F_c = \phi_f \gamma_f + \phi_{rf} \gamma_{rf} (1 - \gamma)_{rf} - F_u, \quad (11)$$

where ϕ_f (volume flux) is assigned by

$$\phi_f = \mathbf{U}_f \cdot \mathbf{S}_f. \quad (12)$$

Before describing each of the quantities appearing in equations (10)–(12), it is helpful to see the role of the delimiter λ_M in the treatment of the liquid fraction advection. This limiter is implemented in the MULES solver (Multidimensional Universal Limiter with Explicit Solution) [70], and is equal to one in the transition region (interface) and zero elsewhere. This is demonstrated in figure 2, where λ_M values are plotted for an arbitrary liquid body translating into the gas phase domain. Away from the interface $\lambda_M = 0$, resulting in

$$(F_u + \lambda_M F_c) = \phi_f \gamma_{f,\text{upwind}}, \quad (13)$$

which treats the advection with a straightforward upwind scheme. Surrounding the gas–liquid interface, $\lambda_M = 1$, and the advection takes the following form:

$$(F_u + \lambda_M F_c) = \phi_f \gamma_f + \phi_{rf} \gamma_{rf} (1 - \gamma)_{rf}. \quad (14)$$

This corresponds to a more sophisticated treatment combining an interfacial compression flux ($\phi_{rf} \gamma_{rf} (1 - \gamma)_{rf}$) with higher order schemes for $\phi_f \gamma_f$. The motivation for this dual treatment is both a reduction of

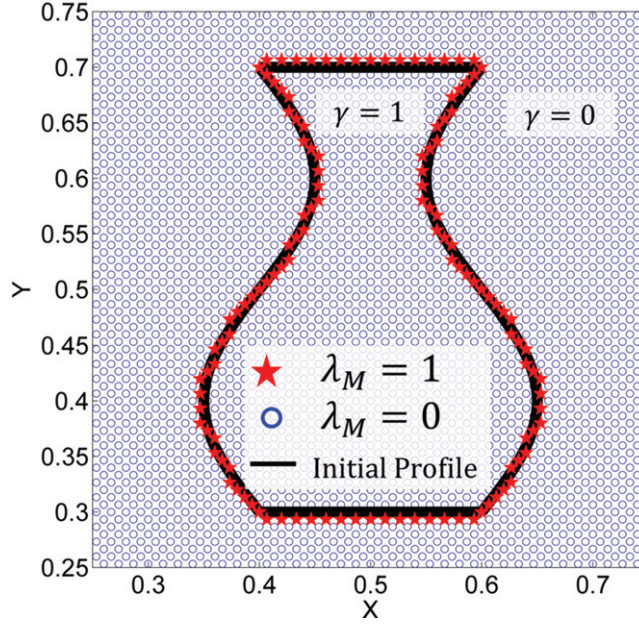


Figure 2. The MULES limiter, λ_M , is equal to one near the interface and to zero away from it.

numerical diffusion at the interface and a reduction of the computational cost by confining the higher order treatment to the interfacial region.

Returning to equations (10)–(12), the cell face area vector \mathbf{S}_f is the product of the unit normal *outward* vector corresponding to cell P and the surface area of the face. The face velocity \mathbf{U}_f in equation (12) is obtained by a general point-to-point interpolation between cells P and N . The `interFoam` code provides linear, cubic and midpoint interpolation options for this operation. The general form for such a point-to-point interpolation is given by

$$\mathbf{U}_f = \lambda_{f,1} \mathbf{U}_P + \lambda_{f,2} \mathbf{U}_N, \quad (15)$$

where $\lambda_{f,1}$, $\lambda_{f,2}$ are the weights for the P and N cells, respectively. These weights are determined by the choice of interpolation as well as by the distance between cell centers. Apart from the general point-to-point interpolation, `interFoam` also offers a variety of interpolation schemes specifically for the divergence (advective) terms (a summary of these schemes can be found in [71]). Additionally, it has the necessary machinery to evaluate interpolations in the context of non-uniform and irregular grids as well.

For the sake of simplifying the code description in both the presentation of the γ transport equation as well as two-phase momentum (section 2.2), we will assume a uniform grid spacing with linear interpolation. This results in $\lambda_{f,1} = 1/2$, $\lambda_{f,2} = 1/2$. Under these conditions

$$\mathbf{U}_f = \frac{\mathbf{U}_P + \mathbf{U}_N}{2}, \quad (16)$$

yielding a second-order spatial convergence. The face values for liquid fraction are calculated using a mixture of central and upwind schemes as follows:

$$\gamma_f = (1 - \lambda_\gamma)(\gamma_f)_{UD} + \lambda_\gamma(\gamma_f)_{CD}, \quad (17)$$

where

$$(\gamma_f)_{UD} = \begin{cases} \gamma_P & \text{for } \phi_f \geq 0, \\ \gamma_N & \text{for } \phi_f < 0 \end{cases} \quad \text{and} \quad (\gamma_f)_{CD} = \frac{\gamma_P + \gamma_N}{2}. \quad (18)$$

The limiter λ_γ can be chosen from among several alternatives [70], which include vanLeer [72], SuperBee [73], Minmod [73], QUICK [74] and GammaDifferencing schemes [75]. In all the results presented here, the vanLeer scheme was used to obtain the face centered values for γ .

Substitution of equation (18) into equation (17) yields the following form for the face centered γ field:

$$\gamma_f = \gamma_P + \frac{(\gamma_N - \gamma_P)}{2} [1 - \varsigma(\phi_f)(1 - \lambda_\gamma)], \quad (19)$$

where $\varsigma(\phi_f)$ is a step function defined by

$$\varsigma(\phi_f) = \begin{cases} 1 & \text{for } \phi_f \geq 0, \\ -1 & \text{for } \phi_f < 0. \end{cases} \quad (20)$$

Regarding the compressive flux, $\phi_{rf}\gamma_{rf}(1 - \gamma)_{rf}$, mentioned previously, the term (ϕ_{rf}) in equation (11) is given by

$$\phi_{rf} = \min \left(C_\gamma \frac{|\phi_f|}{|\mathbf{S}_f|}, \max \left[\frac{|\phi_f|}{|\mathbf{S}_f|} \right] \right) (\mathbf{n}_f \cdot \mathbf{S}_f), \quad (21)$$

where the max operation is performed over the entire domain, while the min operation is done locally at each face $f \in \partial\Omega_i$. The constant C_γ is a user-specified value, which serves as a parameter to restrict interface smearing. In all our simulations, $C_\gamma = 1$ has been used, since increasing or decreasing it was found to exacerbate errors in interfacial curvature and interfacial smearing. The quantity \mathbf{n}_f in equation (21) is the face centered interface normal vector and is given by

$$\mathbf{n}_f = \frac{(\nabla\gamma)_f}{|(\nabla\gamma)_f|} = \left[\frac{(\nabla\gamma)_P + (\nabla\gamma)_N}{2} \right] / \left[\left| \frac{(\nabla\gamma)_P + (\nabla\gamma)_N}{2} \right| \right]. \quad (22)$$

Finally, the quantities γ_{rf} and $(1 - \gamma)_{rf} = (1 - \gamma_{rf})$ in equation (11) are obtained using the `interfaceCompression` scheme [71], shown below:

$$\gamma_{rf} = \gamma_P + \frac{(\gamma_N - \gamma_P)}{2} [1 - \varsigma(\phi_f)(1 - \lambda_{\gamma r})], \quad (23)$$

where the limiter $\lambda_{\gamma r}$ for this scheme (`gammaScheme`) is given by

$$\lambda_{\gamma r} = \min \left\{ \max \left(1 - \max \left[(1 - (4\gamma_P(1 - \gamma_P)))^2, (1 - (4\gamma_N(1 - \gamma_N)))^2 \right], 0 \right), 1 \right\}. \quad (24)$$

2.2. Solution of the momentum equation

In a two-phase flow simulation with fluid densities ρ_l and ρ_g , viscosities μ_l and μ_g and surface tension coefficient σ , the flow is governed by the following momentum equation [76]:

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \otimes \mathbf{U}) = -\nabla p + [\nabla \cdot (\mu \nabla \mathbf{U}) + \nabla \mathbf{U} \cdot \nabla \mu] + \rho \mathbf{g} + \int_{\Gamma} \sigma \kappa \delta(\mathbf{x} - \mathbf{x}_s) \mathbf{n} d\Gamma(\mathbf{x}_s), \quad (25)$$

where Γ is the gas-liquid interface, $\delta(\mathbf{x} - \mathbf{x}_s)$ is the three-dimensional (3D) Dirac delta function and the viscous term $\nabla \cdot (\mu[\nabla \mathbf{U} + \nabla \mathbf{U}^T])$ has been rewritten as $\nabla \cdot (\mu \nabla \mathbf{U}) + \nabla \mathbf{U} \cdot \nabla \mu$. In `interFoam`, the continuum surface force (CSF) model of Brackbill *et al* [28] is employed.

We first perform a volume integral over Ω_i for equation (25). This gives for the surface tension term

$$\int_{\Gamma \cap \Omega_i} \sigma \kappa \mathbf{n} d\Gamma(\mathbf{x}_s) = \int_{\Omega_i} \sigma \kappa \nabla \gamma dV, \quad (26)$$

where $\Gamma \cap \Omega_i$ is the section of the interface that lies imbedded in Ω_i . The association of the surface integral on the lhs with its equivalent on the right-hand side (rhs) is due to the aforementioned CSF model [28]. Furthermore, the local interfacial curvature κ [28, 77] is obtained from the volume fractions using

$$\kappa = -\nabla \cdot \mathbf{n}. \quad (27)$$

The solver employs a modified pressure (p_d) rather than p and their relationship is given by

$$p_d = p - \rho \mathbf{g} \cdot \mathbf{x}, \quad \nabla p_d = \nabla p - \rho \mathbf{g} - \mathbf{g} \cdot \mathbf{x} \nabla \rho. \quad (28)$$

With both the surface tension term and the substitution of p_d for pressure, the volume integral of equation (25) results in

$$\begin{aligned} \int_{\Omega_i} \frac{\partial \rho \mathbf{U}}{\partial t} dV + \int_{\partial \Omega_i} (\rho \mathbf{U} \mathbf{U}) \cdot \mathbf{n} dS = & - \int_{\Omega_i} \nabla p_d dV - \int_{\Omega_i} \mathbf{g} \cdot \mathbf{x} \nabla \rho dV \\ & + \int_{\Omega_i} \sigma \kappa \nabla \gamma dV + \int_{\partial \Omega_i} (\mu \nabla \mathbf{U}) \cdot \mathbf{n} dS + \int_{\Omega_i} \nabla \mathbf{U} \cdot \nabla \mu dV. \end{aligned} \quad (29)$$

The solution of the momentum equation in `interFoam` is performed by constructing a predicted velocity field and then correcting it using the Pressure Implicit with Splitting of Operators (PISO) [78] implicit pressure correction procedure to time advance the pressure (p_d) and velocity fields. The cell in question will be identified by subscript P , as shown in figure 1, and its neighbors by N . In contrast to the previous description for the transport of γ , the details of the momentum algorithm require us to divulge that some of the faces f separating cells P and N are actually owned by N , and the remaining shared faces are owned by P .¹ For the γ transport equation it was not necessary to make this distinction, but for the present momentum case, the iteration level associated with a given variable depends on whether or not the variable pertains to a cell in question.

The PISO iteration procedure is indexed by m , with $m = 0$ corresponding to the initial step and pertaining to the present time level t^n . First, we consider the discrete version of equation (29) accounting only for the temporal, viscous and advective terms and neglecting for the moment pressure, surface tension and gravity terms. This yields an explicit expression for the predicted velocity field \mathbf{U}_p^r , namely

$$\frac{(\rho_P^{n+1} \mathbf{U}_P^r) - (\rho_P^n \mathbf{U}_P^n)}{\Delta t} |\Omega_P| + \sum_{f \in \partial \Omega_i} (\rho_f \phi_f)^n \mathbf{U}_f^{r'} = \sum_{f \in \partial \Omega_i} \mu_f^{n+1} \left(\nabla_f^\perp \mathbf{U} \right)^{r'} |\mathbf{S}_f| + \nabla \mathbf{U}_P^n \cdot \nabla \mu_P^{n+1} |\Omega_P|. \quad (30)$$

In this expression, the fluid density (ρ) and viscosity (μ) are obtained by

$$\rho = \gamma \rho_l + (1 - \gamma) \rho_g, \quad \mu = \gamma \mu_l + (1 - \gamma) \mu_g, \quad (31)$$

and since γ is known at time level t^n and t^{n+1} , the density and viscosity fields are also known.

Prior to a description of the advection and viscous terms, the velocity corresponding to a given face f needs to be defined. It is given by

$$\mathbf{U}_f^{r'} = \frac{\mathbf{U}_P^r}{2} [1 + \Theta(f)w] + \frac{\mathbf{U}_N^m}{2} [1 - \Theta(f)w], \quad (32)$$

where

$$\Theta(f) = \begin{cases} 1 & \text{if } P \text{ is the owner for face } f \text{ and } N \text{ is the neighbor,} \\ -1 & \text{if } N \text{ is the owner for face } f \text{ and } P \text{ is the neighbor} \end{cases} \quad (33)$$

and

$$w = \varsigma(\phi_f) (1 - \lambda_U). \quad (34)$$

In this equation $\varsigma(\phi_f)$ is defined in equation (20) and λ_U carries the same meaning as the limiter function appearing in equation (17). With respect to the viscous term, the surface gradient operator ∇_f^\perp acting on \mathbf{U} is given by

$$(\nabla_f^\perp \mathbf{U})^{r'} = \Theta(f) \frac{\mathbf{U}_N^m - \mathbf{U}_P^r}{|\mathbf{d}|}. \quad (35)$$

Putting equations (35) and (32) into equation (30) yields

$$A_P \mathbf{U}_P^r = \left(\sum_{N} A_N \mathbf{U}_N^m + \mathbf{E}_P^n \right) = H(\mathbf{U}^m) \quad (36)$$

¹ The assignment of owner and neighbor for a given face f is handled within the mesh data structure of the code and by the class `polyMesh`.

after some manipulation. The quantities A_P , A_N and \mathbf{E}_P (not matrices) are given as

$$\mathbf{E}_P^n = \frac{(\rho \mathbf{U})_P^n}{\Delta t} + \nabla \mathbf{U}_P^n \cdot \nabla \mu_P^{n+1}, \quad (37)$$

$$A_P = \left(\frac{\rho_P^{n+1} |\Omega_P|}{\Delta t} + \sum_{f \in \partial \Omega_i} \frac{1}{2} (\rho_f \phi_f)^n [1 + \Theta(f)w] + \sum_{f \in \partial \Omega_i} \mu_f^{n+1} \Theta(f) \frac{|\mathbf{S}_f|}{|\mathbf{d}|} \right) \frac{1}{|\Omega_P|}, \quad (38)$$

$$A_N = \left(-\frac{1}{2} (\rho_f \phi_f)^n [1 - \Theta(f)w] + \mu_f^{n+1} \Theta(f) \frac{|\mathbf{S}_f|}{|\mathbf{d}|} \right) \frac{1}{|\Omega_P|}. \quad (39)$$

Including the effect of surface tension and gravity forces in equation (36), we obtain

$$\mathbf{U}_P^r = \frac{H(\mathbf{U}^m)}{A_P} + \frac{(\sigma \kappa \nabla \gamma)^{n+1}}{A_P} - \frac{(\mathbf{g} \cdot \mathbf{x} \nabla \rho)^{n+1}}{A_P}. \quad (40)$$

Due to the issues with collocated variable arrangement observed in the case of high-density ratio flows [77], equation (40) is solved by interpolating \mathbf{U}_P^r to the cell faces, yielding

$$(\mathbf{U}_P^r)_f = \left(\frac{H(\mathbf{U}^m)}{A_P} \right)_f + \left(\frac{(\sigma \kappa \nabla \gamma)^{n+1}}{A_P} \right)_f - \left(\frac{(\mathbf{g} \cdot \mathbf{x} \nabla \rho)^{n+1}}{A_P} \right)_f. \quad (41)$$

The associated volume flux ($\phi_f^r = (\mathbf{U}_P^r)_f \cdot \mathbf{S}_f$) is obtained by performing an inner product with face surface area, giving

$$\phi_f^r = \left(\frac{H(\mathbf{U}^m)}{A_P} \right)_f \cdot \mathbf{S}_f + \left(\left(\frac{1}{A_P} \right)_f (\sigma \kappa)_f^{n+1} (\nabla_f^\perp \gamma)^{n+1} \right) |\mathbf{S}_f| - \left(\left(\frac{1}{A_P} \right)_f (\mathbf{g} \cdot \mathbf{x})_f^{n+1} (\nabla_f^\perp \rho)^{n+1} |\mathbf{S}_f| \right), \quad (42)$$

where the surface gradient operator (equation (35)) has been used, e.g. $(\nabla \gamma)_f^{n+1} \cdot \mathbf{S}_f = (\nabla_f^\perp \gamma)^{n+1} |\mathbf{S}_f|$.

Equation (42) completes the predictor step. The pressure contribution in terms of a flux,

$$\left(\frac{-\nabla p_d}{A_P} \right)_f \cdot \mathbf{S}_f = \left(\frac{-1}{A_P} \right)_f \left(\nabla_f^\perp p_d^{m+1} \right) |\mathbf{S}_f|, \quad (43)$$

is now added to equation (42) to yield

$$\begin{aligned} \phi_f^{m+1} = & \left(\frac{H(\mathbf{U}^m)}{A_P} \right)_f \cdot \mathbf{S}_f + \left(\left(\frac{1}{A_P} \right)_f (\sigma \kappa)_f^{n+1} (\nabla_f^\perp \gamma)^{n+1} \right) |\mathbf{S}_f| - \left(\left(\frac{1}{A_P} \right)_f (\mathbf{g} \cdot \mathbf{x})_f^{n+1} (\nabla_f^\perp \rho)^{n+1} |\mathbf{S}_f| \right) \\ & - \left(\frac{1}{A_P} \right)_f \left(\nabla_f^\perp p_d^{m+1} \right) |\mathbf{S}_f|. \end{aligned} \quad (44)$$

We have used the superscript $m+1$ above, because p_d is being treated implicitly. The above equation can be rewritten as

$$\phi_f^{m+1} = \phi_f^r - \left(\frac{1}{A_P} \right)_f \left(\nabla_f^\perp p_d^{m+1} \right) |\mathbf{S}_f|. \quad (45)$$

To determine the unknown pressure field p_d^{m+1} , we impose mass conservation for an incompressible medium, i.e. $\sum_{f \in \partial \Omega_i} \phi_f^{m+1} = 0$. This results in

$$\sum_{f \in \partial \Omega_i} \left(\frac{1}{A_P} \right)_f \left(\nabla_f^\perp p_d^{m+1} \right) |\mathbf{S}_f| = \sum_{f \in \partial \Omega_i} \phi_f^r. \quad (46)$$

Equation (46) results in a linear system for p_d^{m+1} , and can be solved using the preconditioned conjugate gradient (PCG) method. Besides PCG, OpenFOAM provides various other options [70] such as preconditioned bi-conjugate gradient, generalized geometric-algebraic multi-grid and `smoothSolver`, which uses a smoother

for convergence. After p_d^{m+1} is obtained using equation (46), we compute ϕ_f^{m+1} using equation (45) for each face. The cell centered velocity fields \mathbf{U}_P^{m+1} are obtained by reconstructing the face velocity flux using the following expression:

$$\mathbf{U}_P^{m+1} = \mathbf{U}_P^r + \left(\frac{1}{A_P} \right) \left(\sum_{f \in \partial \Omega_i} \frac{(\mathbf{S}_f \otimes \mathbf{S}_f)}{|\mathbf{S}_f|} \right)^{-1} \cdot \left(\sum_{f \in \partial \Omega_i} \left(\frac{\phi_f^{m+1} - (\mathbf{U}_P^r)_f \cdot \mathbf{S}_f}{\left(\frac{1}{A_P} \right)_f} \right) \frac{\mathbf{S}_f}{|\mathbf{S}_f|} \right), \quad (47)$$

which marks the end of a single PISO iteration loop. The next PISO iteration is performed by changing the superscript m to $m + 1$ and updating $H(\mathbf{U}^m)$ in equation (36) with the velocity obtained from equation (47), i.e. \mathbf{U}^{m+1} , and solving the equations through equation (47), thereby updating p_d , ϕ_f and \mathbf{U} . This procedure, i.e. from equations (36) to (47), is repeated M times to ensure that the velocity and pressure field together satisfy both continuity and momentum equations. In the computations reported here, three PISO iteration levels (i.e. $M = 3$) were used. At the end of the iteration procedure, the values corresponding to the M th iteration are assigned to the new time step $(n + 1)$, which marks the end of the solution procedure for the t^{n+1} time level, namely

$$\phi_f^{n+1} = \phi_f^M, \quad \mathbf{U}^{n+1} = \mathbf{U}^M \quad \text{and} \quad p_d^{n+1} = p_d^M.$$

3. Performance tests on interFoam

In this section, we report the verification and validation of *interFoam* based on the key issues discussed in section 1. The configurations to be adopted in these tests follow directly from the respective references, including dimensions and velocity fields.

3.1. Kinematics tests

For these tests, *interFoam* was adapted to bypass the solution of the momentum equation and only the advection of γ was computed using equation (9) with a prescribed velocity field. These types of verification tests have become standard in the evaluation of new interface capturing schemes, e.g. [30, 33, 36, 46, 48, 50, 55, 56, 79–82]. The criteria used in evaluating the performance of such methods typically use various error norms defined over the entire domain or restricted to the interfacial region. These tests are classified as verification exercises.

3.1.1. Notched disc in rotating flow. Solid body rotation of a notched disc (figure 3) is a test commonly used for evaluating the advection capabilities of a flow solver. It was first introduced by Zalesak [50] and variants of that configuration have been used by several researchers.

The performance of *interFoam* was first compared with the different VoF formulations presented by Gopala and van Wachem [79], which employed three different algebraic methods for maintaining a sharp interface, namely the CICSAM [18], FCT [83] and InterGamma schemes [84]. As described previously, *interFoam* also employs an algebraic approach for a sharp interface and therefore a direct comparison with the results of [79] is instructive.

For this particular test, the disc radius is $R = 0.5$, the notch width is $W = 0.12$ and the notch height is $H = 0.5$. The disc is initially located at $(x_0, y_0) = (2.0, 2.75)$ in a domain 4×4 . The domain is discretized using a uniform rectangular mesh of size 200×200 , resolving the notch in 6 cells. The rotation velocity is given by

$$u = -0.5(y - y_0) \quad \text{and} \quad v = 0.5(x - x_0). \quad (48)$$

A zero gradient condition is specified for γ at all the domain boundaries. The fractional error defined in [79] is employed here to draw consistent comparisons, namely

$$\mathcal{E}_{ND1} = \frac{\sum_{j=1}^N |\gamma_j^I - \gamma_j^F|}{\sum_{j=1}^N |\gamma_j^I|}, \quad (49)$$

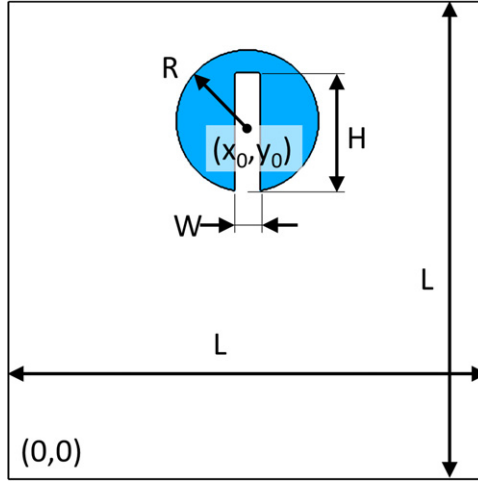


Figure 3. Schematic representation of the notched disc.

Table 2. Comparison of \mathcal{E}_{ND1} errors from *interFoam* with the different VoF solvers [79] corresponding to one rotation of the notched disc.

VoF solvers of [79]			
FCT	CICSAM	InterGamma	<i>interFoam</i>
0.0810	0.0749	0.1218	0.0347

where the summation is carried out over all the \mathcal{N} cells of the domain and superscripts I and F, respectively, refer to the initial and final distributions of γ . In order to ensure that the errors reported here account only for advection and are independent of errors in mapping the analytical solution to the domain, the reference field was chosen to be the one after the first time step, i.e. $\gamma^I = \gamma(t = \Delta t)$. Correspondingly, the final field was chosen to be the one exactly after one rotation. Comparison in table 2 shows that the performance of *interFoam* in this particular test is slightly superior, but in general very much comparable with the implementations of [79].

In addition to the above comparison, *interFoam* was also benchmarked with the following two algorithms, which are not algebraic VoF algorithms,

- (i) the second-order accurate geometric reconstruction scheme (Efficient Least Squares Volume-of-Fluid Interface Reconstruction Algorithm; ELVIRA) of Pilliod and Puckett [80];
- (ii) the Coupled Level Set and Volume of Fluid (CLSVOF) method of Sussman and Puckett [55], which is designed to combine the benefits of level set (accurate curvature) and VoF (mass conservation) methods.

The configuration used for this particular variant of the notched disc test is $R = 0.15$, $W = 0.05$ and $H = 0.25$. The disc is initially positioned at $(0.5, 0.75)$ in a two-dimensional (2D) 1×1 domain and three levels of grid resolution, namely 100×100 , 200×200 and 400×400 , are tested. The following steady velocity field is used to advect the disc for one complete rotation:

$$u = \frac{\pi}{3.14} (0.5 - y) \quad \text{and} \quad v = \frac{\pi}{3.14} (x - 0.5). \quad (50)$$

The error metric, \mathcal{E}_{ND2} , defined below and used in [55, 80], is computed over all the cells in the domain using the initial and final γ values

$$\mathcal{E}_{ND2} = \frac{1}{L} \sum_{j=1}^{\mathcal{N}} A_j \left| \gamma_j^I - \gamma_j^F \right|. \quad (51)$$

Here L is the exact length of the interface and A_j is the area of cell j . The comparison shown in table 3 indicates that the errors incurred in *interFoam* are slightly worse, but of the same order as those obtained

Table 3. Comparison between *interFoam*, CLSVOF [55] and ELVIRA [80] methods in terms of the \mathcal{E}_{ND2} metric.

Method	Grid				
	100 × 100	Rate	200 × 200	Rate	400 × 400
CLSVOF [55]	0.005 72	1.18	0.002 52	1.25	0.001 06
ELVIRA [80]	0.005 67	1.11	0.002 62	1.11	0.001 21
<i>interFoam</i>	0.003 68	1.22	0.001 58	0.72	0.000 96

Table 4. Comparison of \mathcal{E}_{ND1} between linear and cubic interpolation for uniform hexahedral meshes.

Interpolation	Structured grid				
	100 × 100	Rate	200 × 200	Rate	400 × 400
Linear	0.0922	1.234	0.0392	0.726	0.0237
Cubic	0.0922	1.238	0.0391	0.728	0.0236

Table 5. Comparison of \mathcal{E}_{ND1} between linear and cubic interpolation on unstructured meshes.

Interpolation	Unstructured grid				
	100 × 100	Rate	200 × 200	Rate	400 × 400
Linear	0.0787	0.181	0.0694	0.759	0.041
Cubic	0.0787	0.175	0.0697	0.765	0.041

with the CLSVOF and ELVIRA approaches. In addition, the rate of convergence in *interFoam* is comparable to the other methodologies.

It is also instructive to consider how the different interpolation schemes used in computing flux ϕ_f influence the advection results. Therefore, Zalesak's disc was simulated with two interpolation schemes, namely linear and cubic. Relative error, \mathcal{E}_{ND1} with these schemes is reported in table 4 for three different grid resolutions. From these results, the choice of interpolation scheme has a negligible effect on the performance for this test.

In addition, the same test was performed using unstructured meshes having very similar resolutions as the corresponding 100 × 100, 200 × 200 and 400 × 400 meshes. The results are reported in table 5 and show the same trend, i.e. a negligible dependence on the choice of interpolation scheme.

Comparing tables 4 and 5, the results are clearly affected by whether a structured or an unstructured grid is used. The errors with the unstructured grids are somewhat larger compared with the structured grids. Figure 4 shows the final notched disc shape for the 400 × 400 structured and the equivalent unstructured grid. The initial configuration is also shown for reference. While the interface remained relatively smooth for the structured grid, significant ripples were generated in the unstructured case. Such a deformation is especially detrimental when surface tension effects are significant.

3.1.2. A 2D droplet in a vortical flow field. As a second test of the advection algorithm, the stretching and thinning of a 2D object in a vortical flow field is simulated. This test was originally proposed by Rider and Kothe [30] and is aimed at evaluating the capability of the solver in handling severe deformations without causing erroneous numerical breakup.

The test consists of a 2D droplet of radius 0.15 centered at (0.5, 0.75) in a 1 × 1 square domain, subjected to a time reversing velocity field given by

$$\begin{aligned} u &= \sin^2(\pi x) \sin(2\pi y) \cos(\pi t/T), \\ v &= -\sin^2(\pi y) \sin(2\pi x) \cos(\pi t/T). \end{aligned} \quad (52)$$

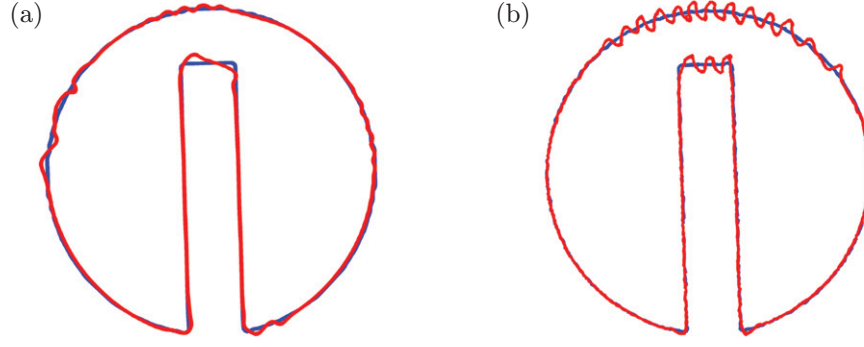


Figure 4. Comparison of the notched disc interface before (blue curve) and after one rotation (red curve) for the 400×400 structured and the equivalent unstructured mesh. (a) A structured grid and (b) an unstructured grid.

Table 6. Comparison of $\mathcal{E}_{\text{RK2D}}$ error with geometrical algorithms [30, 85] and algebraic THINC-SW algorithm [46].

Method	Grid				
	32×32	Rate	64×64	Rate	128×128
PLIC [30]	0.04780	2.78	0.00696	2.27	0.00144
Aulisa <i>et al</i> [33]	0.0253	3.18	0.00278	2.53	0.00048
THINC-SW [46]	0.0390	1.37	0.0152	1.94	0.00396
interFoam	0.1026	1.49	0.0365	0.65	0.0232

This velocity field deforms the droplet into a thin spiraling ligament, which reaches its maximum deformation at $t = T/2$. After this, the flow reverses, restoring the shape at $t = T$. For this test, the period of the flow field is chosen as $T = 8$, following [8, 30, 46]. The advection error is quantified using

$$\mathcal{E}_{\text{RK2D}} = \sum_{j=1}^{\mathcal{N}} A_j \left| \gamma_j^{\text{I}} - \gamma_j^{\text{F}} \right|. \quad (53)$$

The test is performed with three levels of grid resolution, namely 32^2 , 64^2 and 128^2 . The performance of the solver is compared with (i) the geometrical PLIC implementation of Rider and Kothe [30] and (ii) a more recent hybrid method of Aulissa *et al* [33], which uses marker particles along with volume fractions to accurately track the interface. In addition, the performance is also compared with the recently published algebraic algorithm (THINC-SW) of Xiao *et al* [46], which uses a hyperbolic tangent representation for the volume fraction field. Their scheme is reported to significantly reduce the interface smearing.

From table 6, the errors with `interFoam` are of the same order, but consistently three to five times greater than the algebraic THINC-SW scheme and the rate of convergence is correspondingly lower. In comparison with the relatively more complex geometric and hybrid schemes, the performance of `interFoam` is noticeably worse with respect to both error and convergence rates. As a result, the disparity in results becomes even more pronounced at higher grid resolutions.

3.1.3. A 3D droplet in a vortical flow field. As a last test of advection, a 3D extension of the previous test is considered. A sphere of radius 0.15 is subjected to distortion in a 3D time-varying velocity field [30] given by

$$\begin{aligned} u &= 2 \sin^2(\pi x) \sin(2\pi y) \sin(2\pi z) \cos(\pi t/T), \\ v &= -\sin^2(\pi y) \sin(2\pi z) \sin(2\pi x) \cos(\pi t/T), \\ w &= -\sin^2(\pi z) \sin(2\pi x) \sin(2\pi y) \cos(\pi t/T), \end{aligned} \quad (54)$$

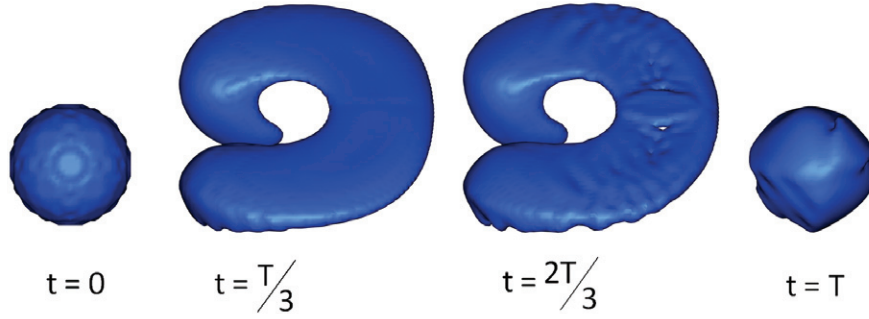


Figure 5. Sphere in a single vortex flow.

Table 7. Comparison of *interFoam* with the THINC-SW [46] algorithm and the FMFPA-3D geometrical reconstruction algorithm [86] in terms of $\mathcal{E}_{\text{RK3D}}$.

Method	Grid					
	32 × 32	Rate	64 × 64	Rate	128 × 128	
THINC-SW [46]	8.39×10^{-3}	1.27	3.47×10^{-3}	1.68	1.08×10^{-3}	
FMFPA-3D [86]	7.44×10^{-3}	1.42	2.79×10^{-3}	1.97	7.14×10^{-4}	
<i>interFoam</i>	9.95×10^{-3}	1.05	4.78×10^{-3}	1.23	2.03×10^{-3}	

with $T = 3$. Figure 5 shows different phases of the deforming sphere. The small undulations at $t = 0$ are the result of visualization of the initial γ field.

The local shape error is quantified using the error norm, $\mathcal{E}_{\text{RK3D}}$, which is defined [46, 86] as

$$\mathcal{E}_{\text{RK3D}} = \sum_{j=1}^{\mathcal{N}} V_j \times \left| \gamma_j^I - \gamma_j^F \right|, \quad (55)$$

where V_j is the cell volume. First, *interFoam* is compared with the algebraic THINC-SW algorithm of [46] and the FMFPA-3D algorithm of [86], which is based on geometric reconstruction using splines. Table 7 shows the comparison. Overall, the errors incurred in *interFoam* are slightly larger, but are still comparable with the THINC-SW scheme.

The errors and convergence rates reported for the geometric algorithm of [86] are somewhat better than the other two schemes. However, despite the accuracy offered by such geometric reconstruction algorithms, due to difficulties involved in their implementation and subsequent computational expenses, algebraic algorithms continue to be used frequently [46]. Nevertheless, the errors incurred in *interFoam* are comparable to the geometric reconstruction algorithm and, additionally, there are no geometrical complexity issues with *interFoam*.

Comparing the behavior of $\mathcal{E}_{\text{RK3D}}$ and analogous $\mathcal{E}_{\text{RK2D}}$ errors with their respective references, 3D calculation appears to have performed slightly better than the 2D counterpart. We hypothesize that this is the result of a significantly more severe distortion in the 2D case as compared to the 3D case.

In addition to comparing the local $\mathcal{E}_{\text{RK3D}}$ error, the percentage mass error was also computed using equation (56). Note that this definition of the error quantifies the difference of the volumes of the droplet before and after advection and is not a local shape error. In table 8 the result is compared with the mass conserving CLSVOF implementation of [81] and the particle level set method of [48].

$$\mathcal{E}_{\%M} = \frac{\sum_{j=1}^{\mathcal{N}} \gamma_j^I - \sum_{j=1}^{\mathcal{N}} \gamma_j^F}{\sum_{j=1}^{\mathcal{N}} \gamma_j^I} \times 100. \quad (56)$$

While the mass error in *interFoam* is extremely small and negligible for practical purposes, it is still not of the order of machine precision as is expected for a mass conserving scheme such as VoF. This is a manifestation of interpolation-related errors introducing a small divergence in the flux representation of the

Table 8. Comparison of $\mathcal{E}_{\%M}$ with different approaches of [48, 81]. All the results are for a $100 \times 100 \times 100$ grid.

Method	% mass error
Level set [81]	32
CLSVOF [81]	0.4
Level set [48]	80
Particle level set [48]	2.6
interFoam	1.303×10^{-7}

velocity field, i.e. $\int_{\partial\Omega_i} \phi_f dS \approx 0$, not exactly equal to 0. This deviation is absent when the full set of equations is solved for velocity and pressure. In that case, ϕ_f is corrected iteratively with pressure in the PISO loop to ensure that continuity is maintained to machine precision.

3.1.4. Local error analysis. The evaluation of global errors for advection test cases, as shown in the preceding section, is a common exercise in the assessment of new interfacial capturing methodologies [30–33, 36, 38, 48–50, 55, 56, 80, 82, 86, 87]. A less common practice is the evaluation of truncation or local errors [88, 89], particularly those related to finite-volume discretization. In the current section, we focus on the advection term of the γ transport equation (equation (7)), since this term is fundamental in correctly capturing and maintaining a relatively sharp material interface. From [88–90] the truncation error for finite-volume discretization is defined as

$$\tilde{\epsilon}_\tau = \frac{1}{|\Omega_i|} \int_{\Omega_i} L \Phi d\Omega - \frac{1}{|\Omega_i|} L^* \Phi^*, \quad (57)$$

where L is the differential operator acting on the exact field Φ , and L^* and Φ^* are the discrete versions of these exact quantities. With respect to the advection term in equation (7), we then have that

$$\tilde{\epsilon}_\tau = \frac{1}{|\Omega_i|} \int_{\partial\Omega_i} \gamma(\mathbf{x}) \mathbf{U}(\mathbf{x}) \cdot \mathbf{n} d\mathbf{S} - \frac{1}{|\Omega_i|} \sum_{f \in \partial\Omega_i} \gamma_f \mathbf{U}_f \cdot \mathbf{n}_f |\mathbf{S}_f|, \quad (58)$$

where the first term on the rhs corresponds to the exact value and the second term to its numerical approximation.

We consider a simple 2D configuration with a uniform velocity field aligned along the x -axis, namely $\mathbf{U}(\mathbf{x}) = 1\mathbf{i}$. The layout of the configuration is displayed in figure 6, where the cell containing the interface is denoted by P and its neighbors by N . The liquid fraction on the left and right neighbors are, respectively, 1 and 0; for the top and bottom neighbors the liquid fraction is the same as cell P , namely $\gamma = 0.2$. The liquid domain is represented by the shaded region in figure 6.

The grid refinement is performed in such a way that the distribution of liquid fraction among cell P and its neighbors N remains unchanged. From equation (58) it follows that

$$\begin{aligned} \tilde{\epsilon}_\tau = \frac{1}{|\Omega_i|} \int_{\partial\Omega_i} \gamma(\mathbf{x}) \mathbf{U}(\mathbf{x}) \cdot \mathbf{n} d\mathbf{S} - \frac{1}{\Delta x^2} \left[\sum_{f \in \partial\Omega_i} \left(\frac{\gamma_P}{2} [1 + \Theta(f) \zeta(\phi_f)(1 - \lambda_\gamma)] \right. \right. \\ \left. \left. + \frac{\gamma_N}{2} [1 - \Theta(f) \zeta(\phi_f)(1 - \lambda_\gamma)] \right) \left(\frac{\mathbf{U}_P + \mathbf{U}_N}{2} \right) \cdot \mathbf{n}_f \Delta x \right], \end{aligned} \quad (59)$$

where a central scheme is used to predict velocities at the faces. The limiter function λ_γ depends on the interpolation scheme used to obtain γ_f . Substituting the values for velocity and γ into equation (59) and performing the exact integral results in the following explicit relation for $\tilde{\epsilon}_\tau$ in terms of λ_γ :

$$\tilde{\epsilon}_\tau \cong -\frac{1}{\Delta x} (0.2 + 0.3\lambda_\gamma). \quad (60)$$

This shows that to leading order the error $\tilde{\epsilon}_\tau$ is proportional to Δx^{-1} independent of the limiter.

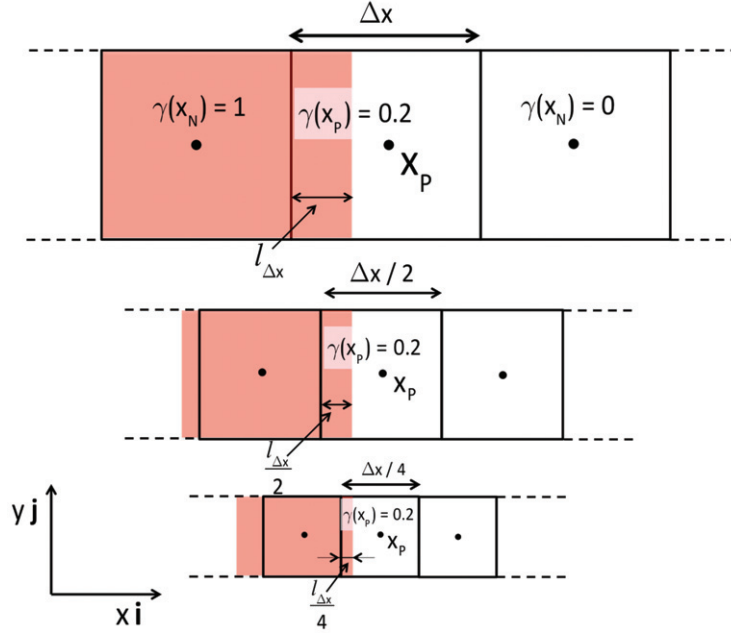


Figure 6. Layout of the volume fraction field for the truncation error test. From top to bottom the grid is refined and the associated distribution of liquid fraction remains unaltered with grid refinement. The colored region corresponds to the liquid.

Table 9. The local error convergence rate (equation (61)) corresponding to the volume fraction field of figure 6 and $\mathbf{U}(\mathbf{x}) = 1\mathbf{i}$.

Grid	$\frac{1}{\Delta x} \int_{\partial\Omega_i} \gamma(\mathbf{x}) \mathbf{U}(\mathbf{x}) \cdot \mathbf{n} d\mathbf{S}$	$\frac{1}{ \Omega_i } \sum_{f \in \partial\Omega_i} (F_u + \lambda_M F_c)$	\tilde{e}_τ	Rate
10×10	-1.00×10	-9.60	-4.00×10^{-1}	
20×20	-2.00×10	-1.92×10	-8.00×10^{-1}	-1.00
50×50	-5.00×10	-4.80×10	-2.00	-1.00
100×100	-1.00×10^2	-9.60×10	-4.00	-1.00

If instead of the Total Variation Diminishing (TVD) discretization of the advection term, the compressive flux as shown in equation (9) is substituted for the truncation term, we obtain

$$\tilde{e}_\tau = \frac{1}{|\Omega_i|} \int_{\partial\Omega_i} \gamma(\mathbf{x}) \mathbf{U}(\mathbf{x}) \cdot \mathbf{n} d\mathbf{S} - \frac{1}{|\Omega_i|} \sum_{f \in \partial\Omega_i} (F_u + \lambda_M F_c). \quad (61)$$

The results are shown in table 9 and indicate that even with the compressive flux substitution, the error behaves as proportional to Δx^{-1} . Curiously, as presented in section 3, the global error in terms of various metrics *does* show sign of convergence. The disparity between the truncation error and these global metrics has been discussed in the literature [89] with respect to finite-volume discretization on irregular grids. The present disparity in the advection of a liquid fraction field represents ground for future research.

3.2. Dynamic tests for inertia-dominated flows

In order to test the solver's ability to correctly simulate both the kinematics and the dynamics of two-phase flow problems, we first consider the flows where inertia dominates the surface tension effects. A number of test cases in this category have already been presented in the literature for *interFoam* (see [3, 8, 9, 12, 17, 18, 41, 91]). Therefore, here we will provide only the tests that have not been previously reported, namely a 2D gravity standing wave and droplet impact and crown propagation. Computational results are compared with experimental and/or analytical findings from the literature. Here both velocity and interface displacement are

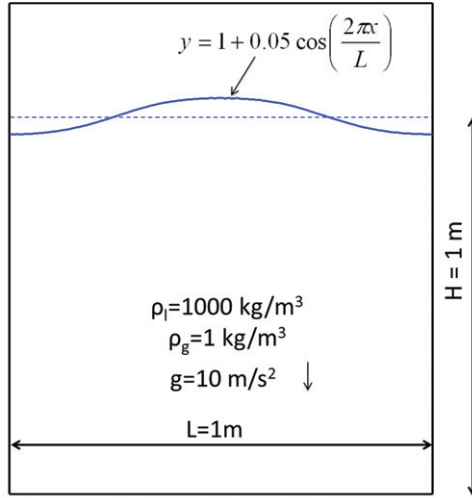


Figure 7. The standing wave case setup.

obtained from the solution of the momentum (equation (25)) and γ transport (equation (7)), i.e. the velocity field is calculated and not imposed as in the previous section.

3.2.1. The standing gravity wave. The case setup considered for the simulation of the 2D standing gravity wave is shown in figure 7. It consists of a fixed mass of liquid bounded by walls on the sides and the bottom. Initial condition is imposed in the form of a perturbed free surface with a profile given by

$$y = H + A \cos\left(\frac{2\pi x}{L}\right), \quad (62)$$

where A is the wave amplitude, H is the average depth of the container and λ is the wavelength of the standing wave. The domain walls are assigned the slip boundary condition to avoid generation of vorticity, as required by the underlying potential flow theory. The analytical solution for the frequency of the wave is given by Lamb [92] as

$$\omega_\lambda = \sqrt{\frac{2\pi g}{\lambda} \tanh\left(2\pi \frac{H}{\lambda}\right)} \approx 7.92 \text{ c/s}. \quad (63)$$

Note that in this formulation, the effects of surface tension and viscosity are neglected. The applicability of the analysis is justified from the values of dimensionless parameters, $We_A = \rho_l(gA)A/\sigma \sim \mathcal{O}(10^2)$; $Re_A = A\sqrt{gA}/\nu \sim \mathcal{O}(10^4)$.

We would like to emphasize here that the analytical solution for the gravity standing wave is based on linearized equations of potential flow. The linearization comes in the form of assumption of small-amplitude oscillations, i.e. $A/H \ll 1$ and $A/\lambda \ll 1$, which allows simplification of kinematic and dynamic boundary conditions at the interface. In our simulations, we specified $A = 0.05\lambda$, which, although small, is not infinitesimal. This is required for ensuring that the amplitude is sufficiently resolved even by the coarsest mesh, but is expected to introduce a small level of nonlinearity to the solution as a secondary effect. The simulations were performed on three levels of resolution, $\lambda/\Delta x = 50, 100$ and 500 . In addition, to demonstrate that surface tension indeed has a negligible influence on the setup considered, another simulation was performed on the finest grid with surface tension.

In order to find the wave amplitude, we obtained the γ profile along the centerline of the domain using the `sample` utility of OpenFOAM. The location of the lowermost cell along the centerline, where γ is closest to 0.5 , is then found using a MATLAB script. This represents the position of the wave crest (or trough) for the particular time with an uncertainty $\mathcal{O}(\Delta x)$. This procedure was repeated at several different times separated by approximately 0.08 s, giving an adequate resolution of ten samples per wave oscillation.

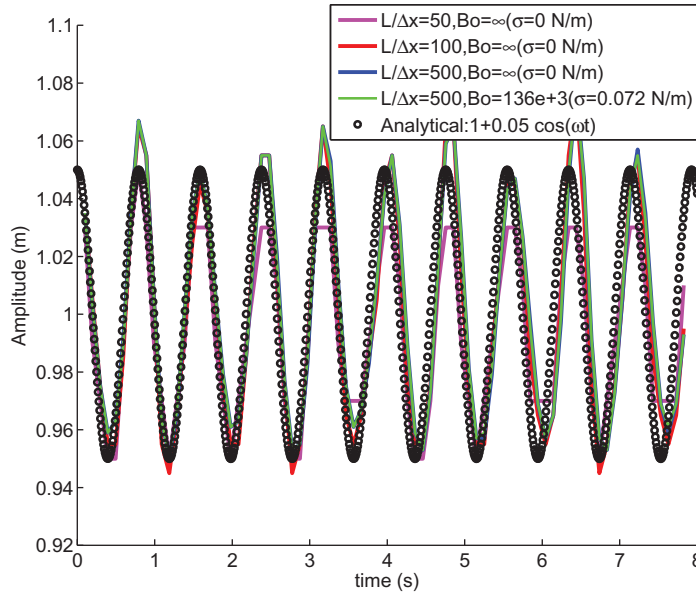


Figure 8. Comparison of the wave front location with the analytical prediction.

As previously expected, a small nonlinearity was apparent from the distorted sinusoidal waveform. This resulted in small overshoots in the wave amplitude, as seen in figure 8. Such a behavior was also observed in the results of [93] based on the CLSVOF method. Nevertheless, this is only a secondary effect and the computational results are in excellent agreement with the analytical frequency.

3.2.2. Droplet impact and crown propagation. The classical single droplet impingement problem has been investigated for over 100 years and continues to attract attention on various aspects of the splashing event. For our purposes, we focus on the liquid crown propagation [94], which provides a clear metric for validating *interFoam*. A droplet of diameter $D = 3.82$ mm with speed $|\mathbf{U}| = 3.56$ m s⁻¹ was allowed to impact a pre-existing liquid film of thickness $T = 2.3$ mm. This configuration corresponds to a Weber number of $We_D \approx 670$ and $Re_D \approx 14 \times 10^3$. A 3D configuration was simulated on a uniform grid of $D/\Delta x = 48$, with a total grid size of about 9 million cells. Results consisting of snapshots of the impingement process are shown in figure 9.

Quantitatively, the predictions of liquid crown diameter are compared with the measurement from [94] in terms of D_{out} and D_{uc} . These diameters are defined pictorially in figure 10(a). The excellent agreement with the experiments demonstrates the ability of *interFoam* to successfully simulate the underlying physics for these types of problems.

Note that Berberovic *et al* [7] have reported similar simulations using *interFoam*, with Reynolds numbers about one order of magnitude smaller than that considered here. Their experimental comparisons extend even into the receding phase of the crater and yield good overall agreement.

3.3. Tests for surface tension-dominated flows

With the solver tested for advection and dynamics for inertia-dominated flows, we shift our attention to low Weber number flows. In the framework of interface capturing methods, the incorporation of surface tension is typically done using the CSF method of [28]. In this framework, surface tension is treated as a body force acting only on the interfacial cells. Three key issues relevant to this treatment are as follows:

- discrete coupling between pressure gradient and surface tension [1],
- computation of interfacial curvature [27],
- spurious currents.

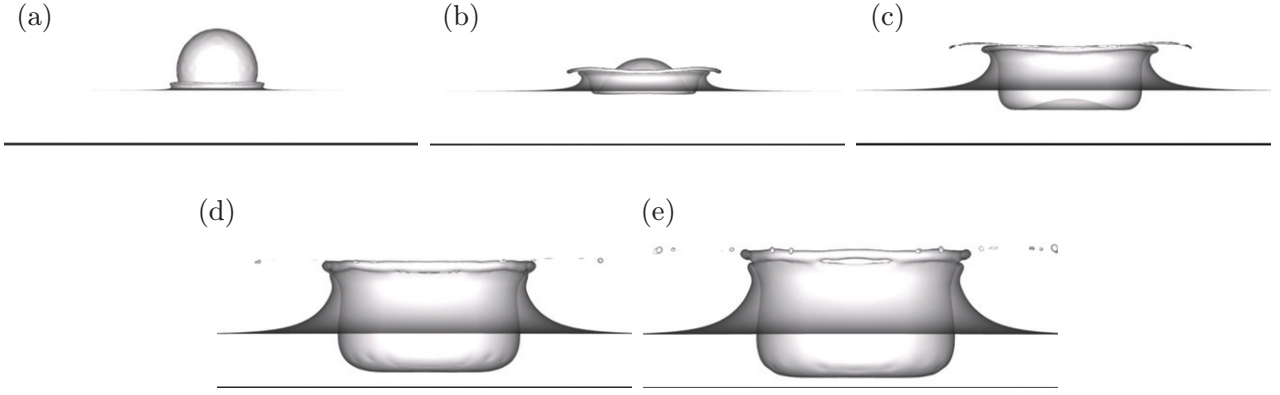


Figure 9. Droplet impact and crown propagation. (a) $t = 0.0004$ s, (b) $t = 0.0008$ s, (c) $t = 0.0015$ s, (d) $t = 0.0025$ s and (e) $t = 0.0030$ s.

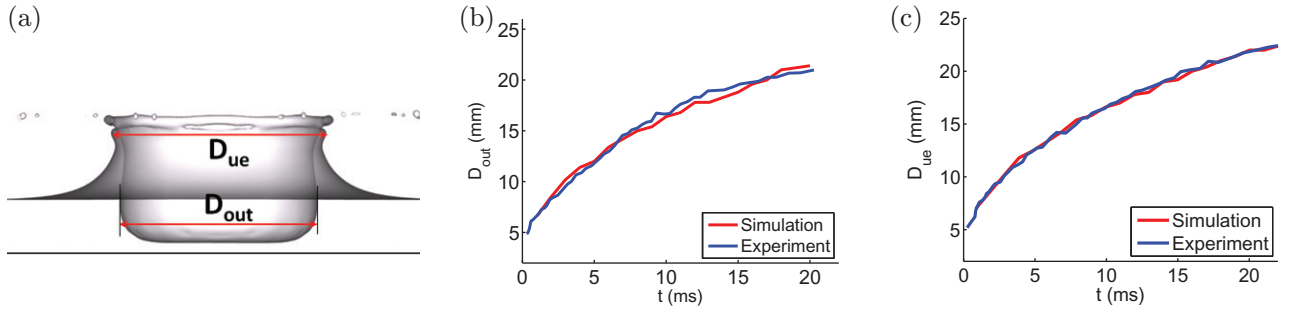


Figure 10. Comparison of *interFoam* with the experiments of Cossali *et al* [95]. (a) Definition of D_{out} and D_{ue} . (b) D_{out} versus t . (c) D_{ue} versus t .

The first two issues are independent of each other, while the third issue is affected by both of them. We will investigate the performance of *interFoam* on all the three aspects making comparisons to the relevant results from the literature.

3.3.1. Test on the coupling between surface tension and pressure. The work of Francois *et al* [1] has established that a consistent coupling between surface tension and pressure is necessary for obtaining a discrete force balance, a required property in surface-tension-dominated flows. Similar to *interFoam*, their algorithm employs a collocated grid arrangement and therefore a comparison of these approaches is instructive.

To evaluate the performance of *interFoam* with respect to the aforementioned discrete balance, we consider an initially static 2D droplet and compute the pressure jump across the interface. For this configuration, a discrete balance implies that the surface tension is balanced exactly by the pressure jump, resulting in no fluid motion. First, to demonstrate that this property is indeed built into the algorithm of *interFoam*, we consider the construction of the pressure Poisson equation (46), where ϕ_f^r contains contributions from the advective, viscous, gravitational and capillary momentum sources (equation (42)). Since we are currently interested in the coupling between pressure and surface tension, we do not consider viscous and gravitational effects and, as a result, ϕ_f^r consists only of the surface tension contribution,

$$\phi_f^r(\mathbf{U} = \mathbf{0}; \mathbf{g} = \mathbf{0}; \mu = 0) = \left(\left(\frac{1}{A_p} \right)_f (\sigma \kappa)_f^{n+1} \nabla^\perp \gamma^{n+1} \right) |\mathbf{S}_f|. \quad (64)$$

In addition, the curvature is constant for the case of a 2D droplet, which simplifies equation (46) to

$$\sum_{f \in \partial \Omega_i} \left(\left(\frac{1}{A_p} \right)_f (\nabla^\perp p_d^{m+1})_f \right) |\mathbf{S}_f| = \sum_{f \in \partial \Omega_i} (\sigma \kappa)_f \left(\left(\frac{1}{A_p} \right)_f^{n+1} (\nabla^\perp \gamma^{n+1})_f \right) |\mathbf{S}_f|. \quad (65)$$

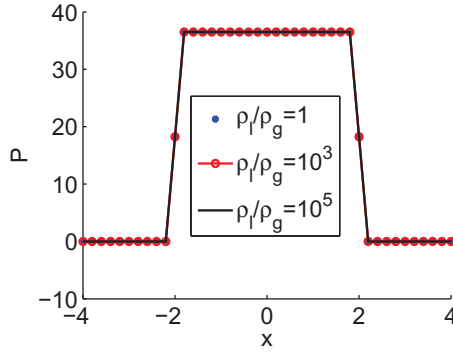


Figure 11. Pressure distribution along the horizontal diagonal of the droplet for different density ratios. The curvature was explicitly specified as $\kappa = 0.5$.

For collocated grids, the pressure gradients must be computed at cell faces and the ∇^\perp operator does precisely this, as shown in section 2. With regard to surface tension, the operator ∇^\perp also operates on γ^{n+1} in the same fashion, resulting in a consistent discrete balance between pressure and surface tension.

We continue to use the 2D static droplet configuration to quantitatively evaluate the performance of *interFoam*. The pressure jump across the interface is given by the Laplace–Young equation

$$\Delta P_{\text{exact}} = \sigma \kappa. \quad (66)$$

The case configuration is taken from [1] and consists of a droplet of radius $R = 2$ in an 8×8 domain with a cell size, $\Delta x = R/10$. Three different density ratios, namely 1, 10^3 and 10^5 , are simulated by keeping the drop density fixed at $\rho_1 = 1$ and changing the density of the surrounding medium accordingly. For all the simulations, $\sigma = 73$ and $v_1 = v_2 = 0$. Since we are only interested in evaluating the discretization of pressure gradient and surface tension at the moment, we exclude the curvature errors by imposing a uniform $\kappa = 1/R$, rather than computing it from γ . We will address curvature predictions in the following subsection.

The computed pressure after one time step, $\Delta t = 10^{-6}$, is shown in figure 11. The pressure jump across the interface takes the correct value $\sigma \kappa = 36.5$, independent of the density ratio. This is the expected behavior because the density ratio does not enter into the Laplace–Young equation.

Further quantification of the performance is provided by comparing different error norms for pressure jump defined by Francois *et al* [1] and magnitudes of spurious currents. The metrics defined there are reproduced here for completeness:

- (i) $\Delta P_{\text{total}} = P_{\text{in}} - P_{\text{out}}$, where the subscripts ‘in’ denotes inside the drop (averaged for cells with $r \leq R$) and ‘out’ outside the drop (averaged for cells with $r > R$),
- (ii) $\Delta P_{\text{partial}} = P_{\text{in}} - P_{\text{out}}$, where the subscripts ‘in’ denotes inside the drop (averaged for cells with $r \leq R/2$) and ‘out’ outside the drop (averaged for cells with $r > 3R/2$) to avoid considering the transition region,
- (iii) $\Delta P_{\text{max}} = P_{\text{max}} - P_{\text{min}}$, where the subscripts ‘max’ denotes maximum and ‘min’ minimum on the entire domain and
- (iv) $\|\mathbf{U}\|_{\text{max}} = \max(\|\mathbf{U}\|)$.

Table 10 shows a comparison with the different approaches of [1]. The performance of *interFoam* is similar to the face centered (FC) CSF and Sharp Surface tension Force (SSF) approaches. This is understandable considering that *interFoam* also employs an FC formulation. Comparison with the cell centered (CC) CSF demonstrates the superiority of these FC approaches.

CSF-FC, SSF-FC and *interFoam* result in similar $E(\Delta P_{\text{partial}})$ and $E(\Delta P_{\text{max}})$ errors. However, $E(\Delta P_{\text{total}})$ for *interFoam* is about 13 orders of magnitude smaller than CSF-FC. This surprising difference is related to the pressure profile and the definition of this error. The $E(\Delta P_{\text{total}})$ error takes into account the interfacial region as well, unlike the other two norms. Thus large errors are expected when the pressure jump is not sharp.

Table 10. Comparison of *interFoam* with the cell and face centered CSF implementations of [1]. Note: the curvature is specified ($\kappa = 0.5 \text{ m}^{-1}$).

	ρ_1/ρ_2	$\ \mathbf{U}\ _{\max}$	$E(\Delta P_{\text{total}})$	$E(\Delta P_{\text{partial}})$	$E(\Delta P_{\text{max}})$
CSF-CC (cell centered [1])	1	2.28×10^{-5}	7.76×10^{-2}	1.03×10^{-14}	7.79×10^{-14}
	10^3				
	10^5		Did not converge		
CSF-FC (face centered [1])	1	1.25×10^{-18}	2.89×10^{-2}	2.73×10^{-15}	8.72×10^{-14}
	10^3	4.97×10^{-18}	2.89×10^{-2}	3.89×10^{-16}	3.89×10^{-16}
	10^5	5.7×10^{-19}	2.89×10^{-2}	1.95×10^{-16}	7.79×10^{-16}
SSF-FC (face centered [1])	1	5.43×10^{-19}	1.36×10^{-15}	5.84×10^{-15}	1.63×10^{-14}
	10^3	4.44×10^{-18}	1.95×10^{-16}	1.17×10^{-15}	3.11×10^{-15}
	10^5	2.71×10^{-19}	3.89×10^{-16}	3.70×10^{-15}	4.87×10^{-15}
<i>interFoam</i>	1	2.95×10^{-19}	1.75×10^{-15}	1.17×10^{-15}	4.28×10^{-15}
	10^3	2.82×10^{-16}	4.48×10^{-15}	1.36×10^{-15}	3.11×10^{-15}
	10^5	1.10×10^{-14}	1.17×10^{-15}	5.84×10^{-16}	1.75×10^{-15}

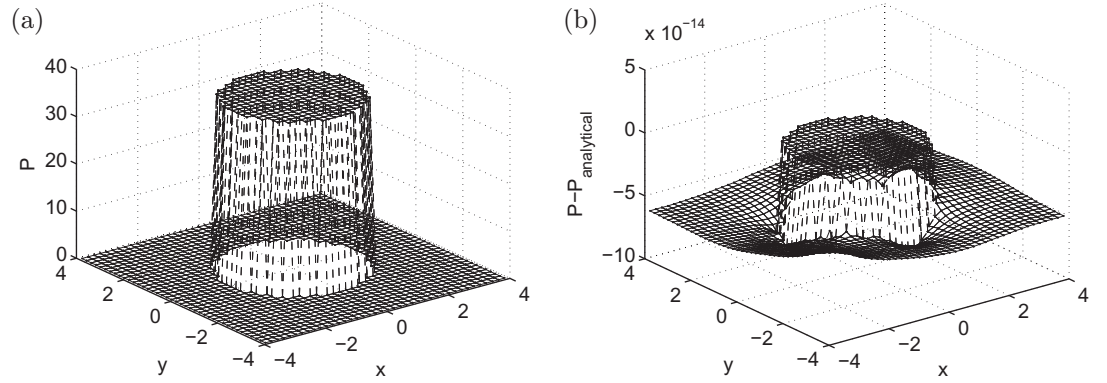


Figure 12. Pressure distribution with *interFoam*: (a) a sharp pressure jump and (b) deviation from theory.

Figure 12(a) shows that the pressure field obtained from *interFoam* is indeed sharp, in contrast to the gradual transition for CSF-FC (see figure 5(a) in [1]). A smoothing operation is applied on the VoF function in CSF-FC which causes the smooth transition in pressure. As a result, intermediate pressure values are generated with CSF-FC, while for *interFoam* an abrupt jump results. Figure 12(b) shows that the deviation of computed pressure from the analytical value is $\mathcal{O}(10^{-14})$.

While the work of [1] revolved around the issue of density ratio, in our computations we observed that the magnitude of spurious currents was directly affected by the magnitude of the gas density, *in addition* to the density ratio. To examine this effect, we consider the momentum balance

$$\frac{\rho D\mathbf{U}}{Dt} = -\nabla p + \nabla \cdot (\mu (\nabla \mathbf{U} + \nabla \mathbf{U}^T)) + \mathbf{F}_\sigma, \quad (67)$$

where \mathbf{F}_σ is the surface tension force. In the absence of viscosity (such as here), the balance reduces to

$$\frac{D\mathbf{U}}{Dt} = \frac{1}{\rho} [-\nabla p + \mathbf{F}_\sigma]. \quad (68)$$

The terms on the rhs of equation (68) represent the net force on the fluid element, ultimately resulting in its (spurious) motion if a perfect balance is not achieved. Note that the factor $1/\rho$ appears in the denominator, which scales the net force acting on the fluid element. With a zero initial velocity (as is the case here), the velocity at the end of the first time step also scales inversely with the density of the fluid element, which is exactly what we observe in our calculations.

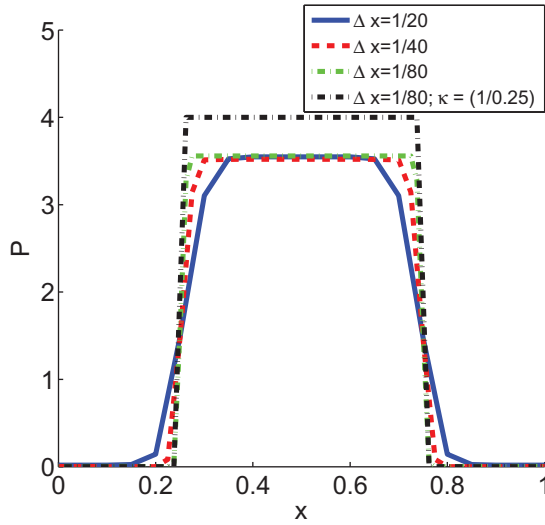


Figure 13. Comparison of pressure inside a 2D globule with the Laplace–Young equation.

While we have shown that there is no imbalance due to inconsistent discretization, an imbalance still exists due to errors associated with the iterative solution of the pressure equation. In this work, we used the PCG solver with the DIC preconditioner. The iteration process continues until the scaled residuals [70] are smaller than the specified tolerance level. In practice, the exact solution is never obtained and the error is related to the solution tolerance introduced. This solution error was confirmed by computing it as follows:

$$\mathcal{R} = \sum_{f \in \partial\Omega_i} \left(\left(\frac{1}{A_p} \right)_f (\nabla^\perp p_d^{m+1})_f \right) - (\sigma\kappa) \sum_{f \in \partial\Omega_i} \left(\left(\frac{1}{A_p} \right)_f^{n+1} (\nabla^\perp \gamma^{n+1})_f \right). \quad (69)$$

For the comparison shown in table 10, a tolerance of 10^{-20} was specified.

3.3.2. Test of interfacial curvature. Now we shift our attention to the second issue concerning interfacial calculations, which is accurate computation of interfacial curvature. In the VoF formulation, it is common to use the volume fractions to compute the interface normal and curvature [51] which, without smoothing or interface reconstruction, provides a stair-stepped approximation to the interface [27]. Although this approximation does not vanish locally with refinement, it is known that the integral effect of curvature (i.e. average pressure jump) converges to a value that is systematically different from the analytical value [32]. This may be a potential problem and therefore we test *interFoam* for the accuracy of curvature estimation.

Again the test case of a 2D stationary droplet in zero force field is used. This time, however, the curvature was not specified—instead, it was computed using equation (27). The pressure jump (and its associated error) is therefore directly related to the curvature (and error in curvature estimation).

For this test, a droplet of $R = 0.25$ centered in a 1×1 square is used. The fluid properties are $\rho_1 = \rho_2 = 10^4$, $\mu_1 = \mu_2 = 1$ and $\sigma = 1$. This setup is the same as that used by Hysing [34]. The computations were run up to $t = 125$ consistent with [34] and the pressure inside the globule at the end of the simulation was compared with that predicted by the Laplace–Young equation ($\Delta P_{\text{exact}} = 4$). This simulation was run on four different grid densities to test the convergence of pressure with grid refinement. The computed pressure along the horizontal diameter of the globule is plotted in figure 13.

For all the grids, the pressure converged to a value $\Delta P_{\text{computed}} \approx 3.6$ rather than 4. The error in pressure estimation ($\mathcal{O}(10\%)$) is almost entirely due to inaccuracies in the computation of interface curvature, confirming the previous findings regarding convergence to a value slightly different from the analytical one.

This systematic error in curvature estimation may be somewhat mitigated through the use of a smoothed VoF field for computing κ , as shown in [1, 35]. Our ongoing work with such smoothing suggests the same trend for *interFoam* as well.

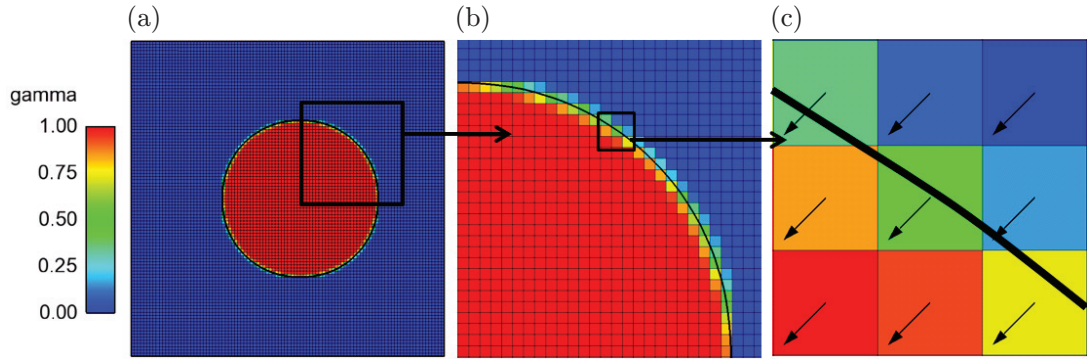


Figure 14. Volume fraction distribution at the interface of a 2D droplet (a). Black curve: position of the true interface. The VoF function varies abruptly (b), resulting in a discrete interface that is poorly represented locally (c).

3.3.3. Spurious currents. Generation of spurious currents, which appear in the form of vortices around the interface, has been reported for several surface tension methods [96] including the CSF method [28, 97]. These flows, also commonly referred to as parasitic currents, are generated even in the absence of any external forcing and are solely due to numerical issues. In the case of the CSF method, the discrete approximation of the interface (figure 14) acts as a perturbation on the physically smooth interface [66] and sets up a spurious capillary flow in its vicinity. The magnitude of this capillary velocity ($C_{\Delta x}$) is related to the lengthscale of these perturbations ($\sim \Delta x$) [28] and is given by

$$C_{\Delta x} \sim \sqrt{\frac{\sigma}{\rho \Delta x}}. \quad (70)$$

From equation (70) it is clear that grid refinement does not mitigate, but instead exacerbates the magnitude of spurious velocity. Apart from the curvature errors, a force imbalance serves to magnify spurious currents through the action of fluid density (equation (68)). It is in this sense that both accurate curvature calculation and discrete balance between pressure gradient and surface tension are required in order to minimize the spurious currents. Once the potential for parasitic flow is established, the explicit treatment of surface tension term places a stability condition on a time step [28, 34], given by

$$\Delta t < \sqrt{\frac{\rho \Delta x^3}{\sigma}}. \quad (71)$$

In a noteworthy contribution by Galusinski and Vigneaux [66], this time step constraint was revisited by including the effect of viscosity, leading to the following generalized time step criterion:

$$\Delta t \leq \tau_\sigma = \frac{1}{2} \left\{ C_2 \frac{\mu \Delta x}{\sigma} + \sqrt{\left(C_2 \frac{\mu \Delta x}{\sigma} \right)^2 + 4 C_1 \frac{\rho \Delta x^3}{\sigma}} \right\}. \quad (72)$$

The performance of *interFoam* within the context of generalized time step requirements equation (72) is evaluated by the following test: a stationary 2D droplet of radius $250 \mu\text{m}$ is initialized in a uniform domain of size $10^{-3} \text{ m} \times 10^{-3} \text{ m}$ and a uniform grid size of $\Delta x = 10^{-5} \text{ m}$. Both the fluids (droplet as well as surroundings) share common fluid properties ρ and μ , which are varied systematically over a set of about 80 simulations. The coefficient of surface tension ($\sigma = 0.01 \text{ N m}^{-1}$) and simulation time step ($\Delta t = 10^{-4} \text{ s}$) are always maintained constant. Gravity is absent in all the simulations.

Spurious currents are always generated initially for the reasons previously explained. However, depending upon the case setup, the flow either becomes unstable or quickly stabilizes. The instability is characterized by the growth of kinetic energy over time leading to displacement of the droplet, as shown in figures 15(a)–(c). Stable configurations, on the other hand, manifest a rapid decay of the spurious flow and consequently no droplet motion/distortion (figures 16(a)–(c)). A distinction between the stable and unstable cases can then be

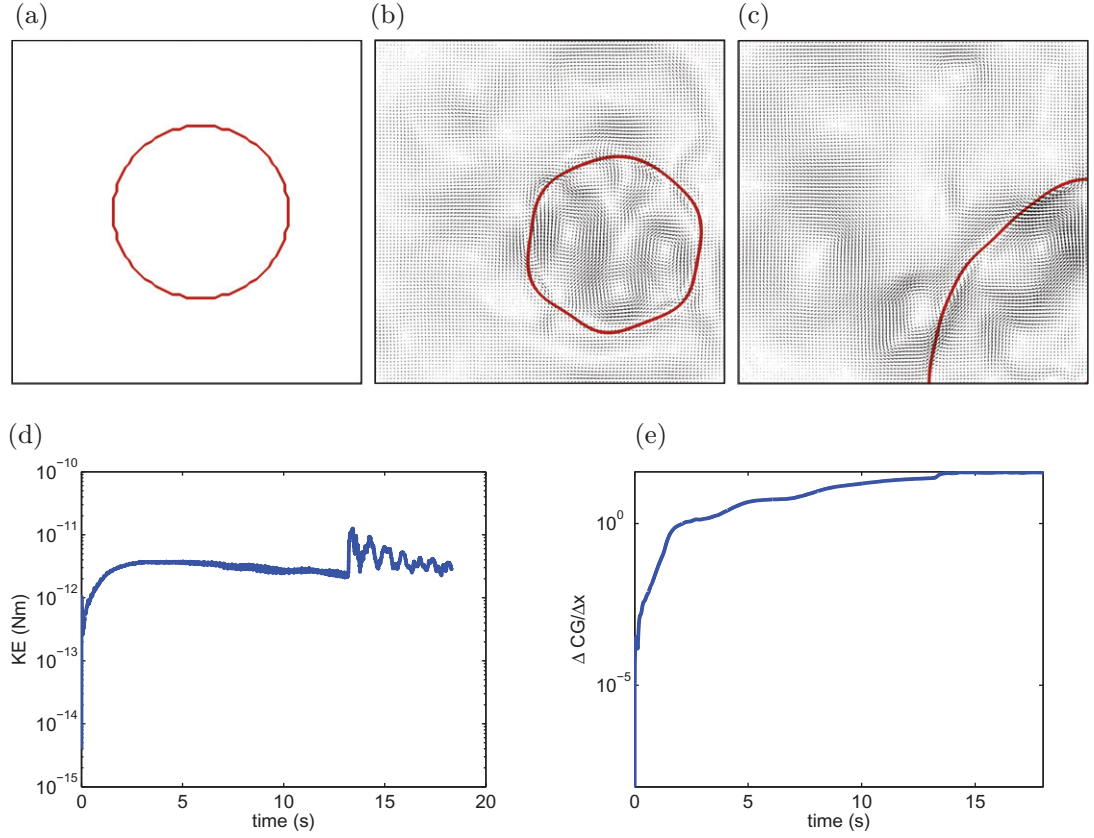


Figure 15. An example of the unstable case. For this case, the kinetic energy and displacement of the droplet continue to increase with time. $\rho = 10^7 \text{ kg m}^{-3}$, $\mu = 10^{-5} \text{ N s m}^{-2}$; $\tau_\rho/\Delta t = 10$, $\tau_\mu/\Delta t = 10^{-4}$. (a) $t = 0 \text{ s}$, (b) $t = 10 \text{ s}$, (c) $t = 20 \text{ s}$. (d) Kinetic energy of the globule. (e) Displacement of the globule.

made on the basis of time history of kinetic energy or displacement of the droplet, which are also plotted in these figures. These quantities are calculated using

$$\text{KE} = \frac{1}{2} \int_{\Omega_1} \rho (\mathbf{U} \cdot \mathbf{U}) dV = \frac{1}{2} \sum_{V_j \in \Omega_1} \rho_j (\mathbf{U} \cdot \mathbf{U}) V_j, \quad (73)$$

$$\Delta \mathbf{x}_c = \|\mathbf{x}_c - \mathbf{x}_{\text{initial}}\|, \quad (74)$$

where $\mathbf{x}_{\text{initial}}$ refers to the location of the center of mass at $t = 0$ and

$$\mathbf{x}_c = \frac{(\int \gamma \mathbf{x} dV)}{(\int \gamma dV)} = \frac{\sum_{j=1}^N (\gamma_j V_j) \mathbf{x}_j}{\sum_{j=1}^N (\gamma_j V_j)}. \quad (75)$$

Here \mathbf{x}_j is the cell center location and V_j is the volume for cell j .

While determining whether the kinetic energy of the droplet grows or decays is a valid criterion for identifying stable and unstable flows, in this work we preferred to make this classification based on the displacement metric. This facilitates a clear differentiation of the stable and unstable cases, given by

$$\begin{aligned} \frac{\Delta \mathbf{x}_{\text{final}}}{\Delta x} &< 1: \text{stable}, \\ \frac{\Delta \mathbf{x}_{\text{final}}}{\Delta x} &> 1: \text{unstable}, \end{aligned} \quad (76)$$

where $\Delta x_{\text{final}} = \Delta x_c(t = t_{\text{end}})$ and Δx is the cell size. The choice of t_{end} is somewhat arbitrary and the only requirement is that it should be large enough for the result (stable/unstable) to be independent

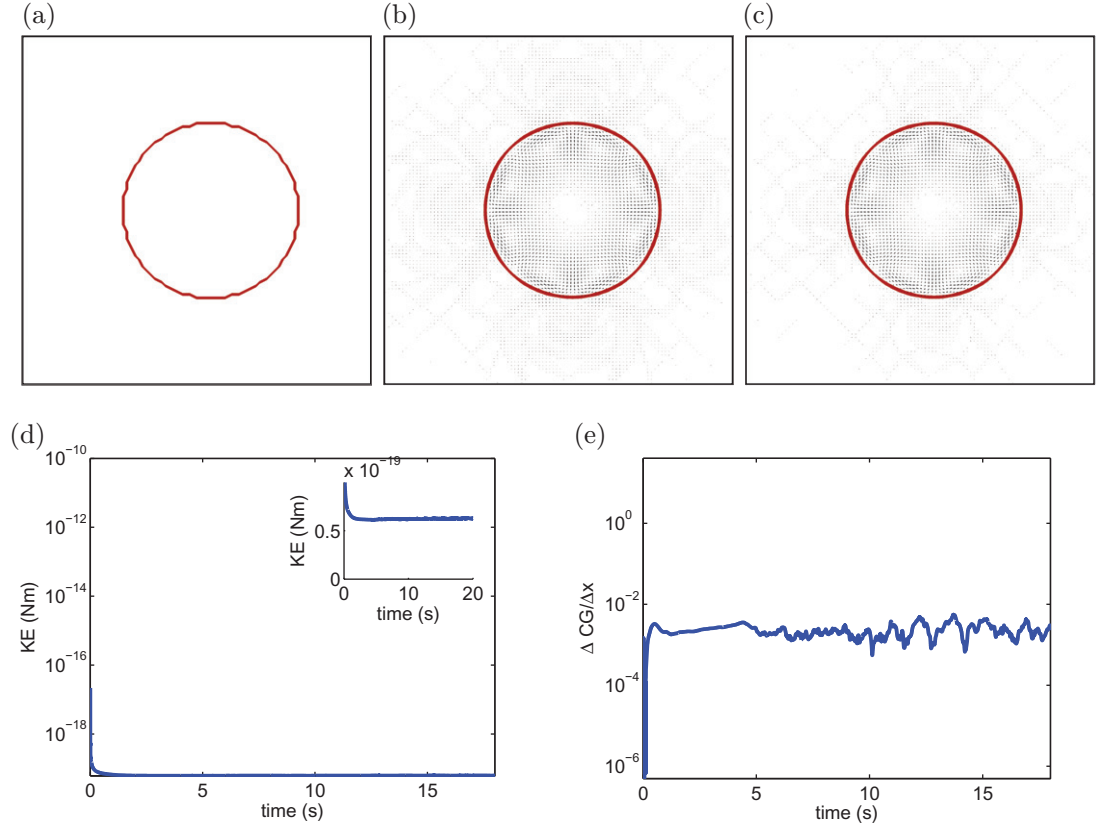


Figure 16. An example of the stable case. For this case, the kinetic energy of the droplet decayed rapidly to negligible values and the droplet remained stationary indefinitely, $\rho = 10 \text{ kg m}^{-3}$, $\mu = 5 \times 10^{-2} \text{ N s m}^{-2}$; $\tau_\rho/\Delta t = 10^{-2}$, $\tau_\mu/\Delta t = 0.5$. (a) $t = 0$ s, (b) $t = 10$ s, (c) $t = 20$ s. (d) Kinetic energy of the globule. (e) Displacement of the globule.

of the assignment of t_{end} . In this work, as documented in the following section, we choose three different times, namely $t_{\text{end}} = 10, 15$ and 20 s. We subsequently checked whether a given computation results in the stable/unstable criteria as defined by equation (76) for all the three times. We found that beyond $t = 10$ s, the qualification of stable/unstable does not change. The only exceptions are for the cases that have an unphysically low viscosity or that have zero viscosity. Under these extreme conditions, the cases are generally unstable. Similar findings in this regime have also been reported in [1, 35].

3.3.4. Control of spurious currents using the analysis of Galusinski and Vigneaux [66]. The time step constraint (equation (72)) obtained from the analysis by Galusinski and Vigneaux [66] consists of two independent time scales

$$\tau_\rho = \sqrt{\frac{\rho \Delta x^3}{\sigma}}, \quad \tau_\mu = \frac{\mu \Delta x}{\sigma}, \quad (77)$$

which depend upon the fluid properties as well as the grid size. Using τ_ρ and τ_μ , equation (72) can be written as

$$\tau_\sigma = \frac{1}{2} \left\{ C_2 \tau_\mu + \sqrt{(C_2 \tau_\mu)^2 + 4 C_1 \tau_\rho^2} \right\}. \quad (78)$$

Note that C_1, C_2 are independent of fluid properties and are only solver specific [66].

To obtain quantitative information on stable time step for `interFoam`, C_1 and C_2 were determined directly from the simulations, each having a different $(\tau_\rho/\Delta t, \tau_\mu/\Delta t)$ coordinate. The classification given in equation (76) was used to differentiate the stable and unstable cases.

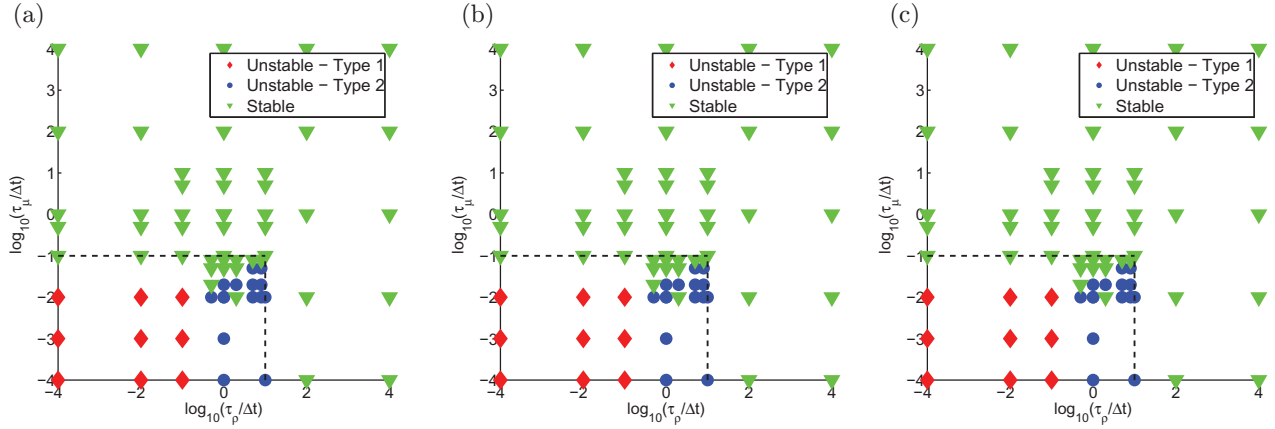


Figure 17. Stability chart for different integration times t_{end} : (a) $t_{\text{end}} = 10$ s, (b) $t_{\text{end}} = 15$ s and (c) $t_{\text{end}} = 20$ s.

The results for $t_{\text{end}} = 5, 15$ and 20 s are shown in figure 17. In these maps, each symbol represents one simulation. The boundary between the stable and unstable computations is approximately indicated by a black dashed line. Three types of results can be seen from the stability plot.

- (i) *Unstable, type-1 (red diamonds)*. For these cases, the generated spurious velocities were too large even for the Courant number criterion (i.e. $Co \gg 1$ for these cases). These cases crashed before completion.
- (ii) *Stable (green triangles)*. For these cases, the spurious velocity and kinetic energy decayed rapidly (typically within $t \sim 10^{-3}$ s) to very small values. The droplet displacement was practically absent ($\mathcal{O}(10^{-3}\Delta x)$).
- (iii) *Unstable, type-2 (blue circles)*: Here the spurious flow did not decay with time but the Courant number restriction was also not violated.

The stability charts at three different times show convergence in the behavior of the stability regime. From these, we obtain the constants $C_1 = 0.01$ and $C_2 = 10$. It has been shown by Harvie *et al* [97] that the generation of spurious currents is only secondary for a moving interface. This was also observed in *interFoam* and therefore the time step analysis shown here represents a conservative estimate.

3.4. Simulation of capillary flows

In this section, the results are presented for some physical situations involving strong surface tension effects.

3.4.1. 2D standing capillary wave. The setup used in this simulation is shown in figure 18. A perturbation with wavelength λ and amplitude $a = \lambda/20$ is allowed to evolve under the influence of surface tension alone. In this case, the appropriate scaling for velocity is $u_{\text{cap}} \sim \sqrt{\sigma\kappa/\rho}$, which gives $We \sim \mathcal{O}(1)$. The fluid properties are provided in the figure. In order to obtain a standing wave, the left and right side boundaries were made cyclic. The kinetic energy of the liquid is calculated as shown in equation (73).

The results are compared with the analytical solution of Lamb [92], which gives the frequency of oscillation in the linear limit (i.e. small-amplitude waves)

$$\omega_{\text{osc}} = \sqrt{\frac{\sigma\kappa^3}{\rho_l + \rho_g}}, \quad \text{where } \kappa = \frac{2\pi}{\lambda}. \quad (79)$$

Analytically, the frequency of change in kinetic energy is twice the frequency of oscillation of the free surface. The analytical period of oscillation of kinetic energy is therefore $\tau_A = 1/2(2\pi/\omega_{\text{osc}}) = 2.384 \times 10^{-5}$ s. The error in the period is then defined by

$$\mathcal{E}_\tau = \frac{\tau - \tau_A}{\tau_A} \times 100. \quad (80)$$

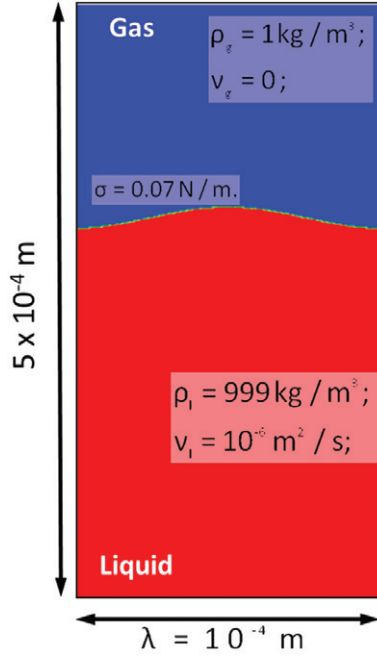


Figure 18. Standing capillary wave setup.

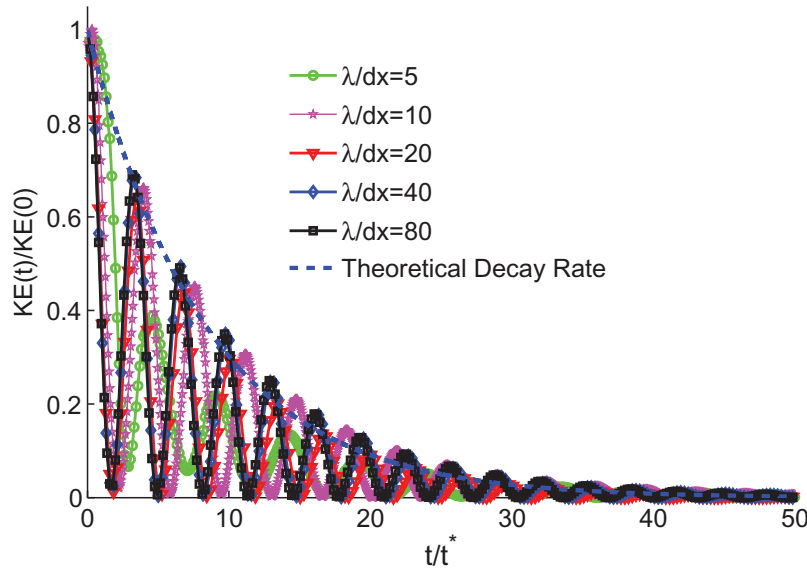


Figure 19. Comparison of the period of kinetic energy for the configuration in figure 18, with the analytical solution [92].

In addition, the rate of decay of kinetic energy due to viscous effects is given by Lamb [92] as

$$\frac{KE(t)}{KE(0)} = e^{-4\nu^* t^*}, \quad \text{where } \nu^* = \nu_l / \kappa^2 \sqrt{(\rho_l / (\sigma \kappa^3))} \quad \text{and} \quad t^* = t / \sqrt{(\rho_l / (\sigma \kappa^3))}. \quad (81)$$

In order to identify the grid requirement for numerical convergence, five different grid densities were tested with $\lambda/\Delta x = 5, 20, 30, 40$ and 80 . The evolution of kinetic energy for these grids is plotted in figure 19 along with the exponential decay of kinetic energy due to viscous effects, calculated from equation (81).

The computed oscillation periods for each grid (averaged over 10 cycles) and the corresponding errors are shown in table 11. The period of oscillation for the coarsest grid is in about 53% error. A significant improvement happens with refinement to $\lambda/\Delta x = 10$. For the next grid ($\lambda/\Delta x = 20$), the error in period with

Table 11. Error and convergence rate of the computed period of kinetic energy for the standing capillary wave of figure 18.

$\lambda/\Delta x$	Calculated period, τ_G (s)	% error, \mathcal{E}_τ	Rate
5	3.657×10^{-5}	53.39	–
10	2.712×10^{-5}	13.76	1.9569
20	2.480×10^{-5}	4.03	1.7739
40	2.440×10^{-5}	2.35	0.7789
80	2.425×10^{-5}	1.72	0.4509

respect to the analytical solution is within 5%. Beyond this, in spite of the inaccuracies in curvature, the results show trends of convergence to a value very close to the theory. This observed loss in rate of convergence can be explained as a combination of systematic error in curvature (section 3.3.2) and also the fact that the analytical solution also contains a systematic error, since it is based on the linearized version of the equations, whereas *interFoam* solves the full version of the Navier–Stokes equations.

3.4.2. Rayleigh breakup of a laminar liquid jet. As a second case, we simulate the Rayleigh breakup of a laminar liquid jet. The problem consists of the growth of a varicose sinusoidal perturbation imposed on the surface of a liquid cylinder with surface tension. The final result is disintegration of the column. This phenomenon was analyzed by Rayleigh [98], considering an inviscid liquid jet in a vacuum. According to the theory, all wavelengths $\lambda_p > 2\pi R$ are unstable and grow in time leading to disintegration of the jet.

However, experimental observations [99] suggested that the theory of Rayleigh was inadequate and nonlinear effects caused formation of satellite droplets in addition to main droplets. Lafrance [100] extended the analysis to include these nonlinearities and provided theoretical estimates for the main and satellite droplet sizes. We used the findings of Lafrance, which are in excellent agreement with their experiments, to test the performance of the solver.

All the simulations are performed on a water jet of $R = 17.5 \times 10^{-6}$ m, with $\rho = 1000 \text{ kg m}^{-3}$, $\mu = 10^{-3} \text{ N s m}^{-2}$ and $\sigma = 0.073 \text{ N m}^{-1}$. The simulations are 3D and are performed using isotropic hexahedral meshes with $\Delta x = 1.3 \times 10^{-6}$ m in the region occupied by the jet and its vicinity. Far away from the jet, coarser cells were used to keep mesh sizes moderate ($\mathcal{O}(1 \text{ million})$). The simulations were performed with a constant time step $\Delta t = 5 \times 10^{-9}$ s, well within the capillary restrictions ($\tau_\mu \approx 2 \times 10^{-8}$ s; $\tau_\rho \approx 2 \times 10^{-7}$ s). Three jets were simulated, each with different perturbation wavelength λ_p given by

$$\frac{\lambda_p}{2\pi R} \approx 1.33, 1.67 \text{ and } 2, \quad (82)$$

with the respective dimensionless wavenumbers $k = (2\pi R)/(\lambda_p) = 0.75, 0.6$ and 0.5 .

Figure 20(a) shows the initial jet with perturbation on it. The perturbation grows in time and finally the jet disintegrates. Figure 20(b) shows the disintegrated jet and identifies the main and satellite droplets. The sizes of the main and satellite droplets are normalized by R and plotted in figure 21. The curves in figure 21 correspond to the theory of Lafrance [100], while the simulations are represented by circles. Red circles correspond to satellite droplets and green to the main droplet. Clearly, the agreement with experimental data is very good. While the main droplet sizes match the theory almost exactly, a slight difference can be seen in computed and predicted satellite droplet sizes. However, looking at the experimental data of [100], this deviation is well within the acceptable limits.

3.4.3. Retraction of a liquid column. Accurate computation of temporal effects of capillary flow becomes an important feature when dealing with flows such as high-speed atomizing jets. The present validation test investigates the accuracy of *interFoam* in capturing these temporal effects in reference to flows relevant to atomization. A qualitative comparison is made with the results presented by Umemura [42].

The configuration simulated here is an initially stationary cylindrical liquid column of SF_6 ($\rho_l = 1460 \text{ kg m}^{-3}$, $\mu_l = 1.1 \times 10^{-3} \text{ N s m}^{-2}$) evolving under the action of surface tension ($\sigma = 1.605 \times 10^{-3} \text{ N m}^{-1}$)

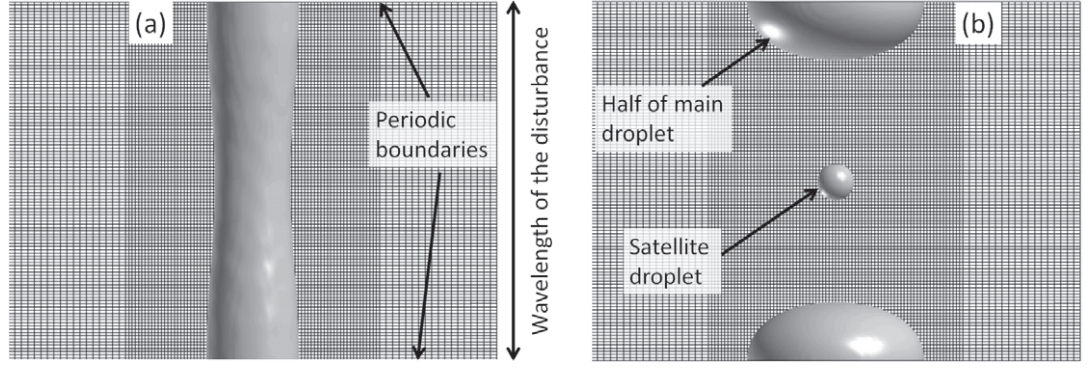


Figure 20. Rayleigh breakup of a water jet ($k = 0.75$). (a) Initial perturbation on the liquid jet. (b) Jet after disintegration results in one satellite and one main droplet.

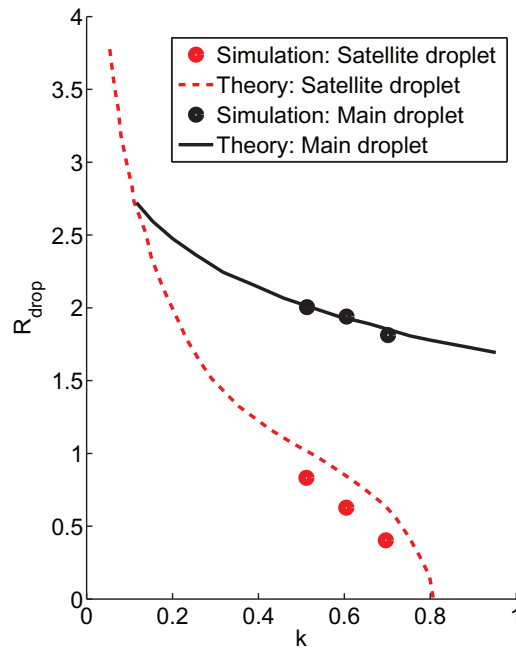


Figure 21. Comparison of simulation results for satellite and main droplet sizes for three different wavenumbers with the corresponding experimental and analytical results of [100].

in a pressurized environment of nitrogen ($\rho_g = 79.1 \text{ kg m}^{-3}$, $\mu_g = 1.76 \times 10^{-5} \text{ N s m}^{-2}$). Note that a density ratio of approximately 20 was used in [42] instead of 1000 (water–air) due to computational difficulties in handling large density ratios. A large density ratio is not a problem for *interFoam*.

The simulation was performed on a uniform mesh with $\Delta x = 7.8 \mu\text{m}$ with 32 grid points across the jet diameter. The total grid size is approximately 7.5 million cells. The simulation was run with a constant time step of $\Delta t = 2 \times 10^{-6} \text{ s}$, which was well within the stable regime with respect to spurious currents ($\tau_\rho \approx 2 \times 10^{-5} \text{ s}$; $\tau_\mu \approx 5 \times 10^{-6} \text{ s}$). Similar to all the other computations reported here, explicit time marching was used for this computation as well.

Figure 22 shows different stages in the evolution of the liquid column. Results are compared pictorially with Umemura's [42] VoF results on a similar grid ($D/\Delta x = 40$) for several non-dimensional times, t^* , given by

$$t^* = tV/a, \quad \text{where } V = \sqrt{\frac{\sigma}{\rho_l a}}. \quad (83)$$

Images on the top correspond to the simulations of Umemura and the ones on the bottom correspond to our simulations.

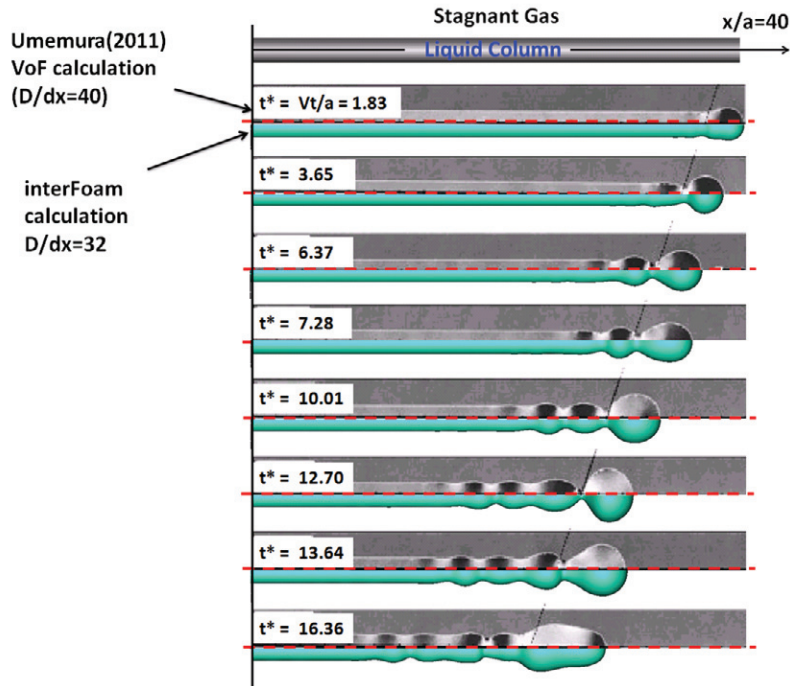


Figure 22. Retraction of a liquid column due to surface tension. Top half: VoF simulation of [42] with $D/\Delta x = 40$; bottom half: VoF simulation using *interFoam* $D/\Delta x = 32$. Reprinted with permission from A. Umemura [42]: Phys. Rev. E 83 pg. 046307-4, 2011. Copyright (2011) by the American Physical Society.

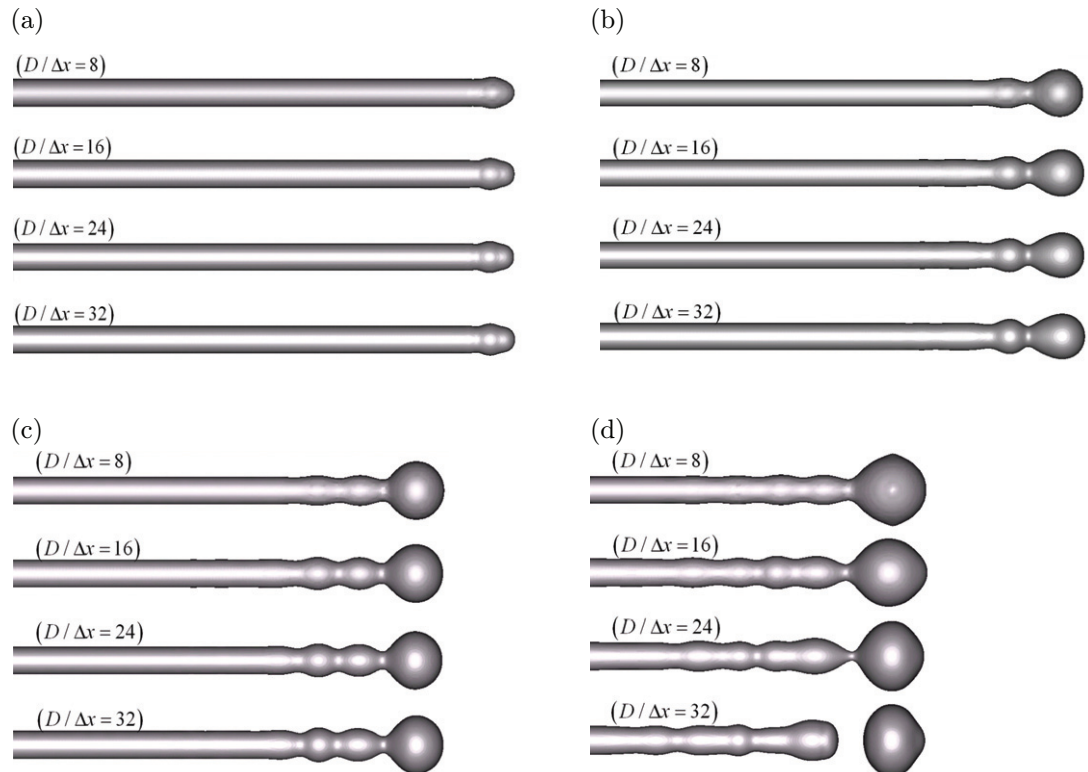


Figure 23. Simulation of the retracting jet on different levels of grid resolution.

Physically, surface tension results in a rapid bulging of the free end of the column, accompanied by retraction and necking. These aspects are captured very accurately with the present solver, resulting in very good overall agreement with the VoF results of [42].

In addition, a grid dependence study was also performed involving the resolutions: $D/\Delta x = 8, 16, 24$ and 32. The results at different non-dimensional times ($t^* = 1, 6.4, 10$ and 19) are shown in figure 23. It is clear that the initial stages of the jet development such as retraction speed and overall jet shape are very well captured by all the grids until about $t^* = 10$. After this, slight differences begin to appear in the amplitudes of waves, which are resolved to differing degree by the four grids. Finally, at $t^* = 19$, the finest grid results in the breakup of the jet, while the other grids do not. It should be noted, however, that this grid dependence is expected and is also noted in the literature [45]. As the jet retracts, the neck diameter shrinks and approaches the cell size. In the framework of interface capturing methods, the interface location is known only approximately and therefore the behavior of any liquid structure of the order of cell size is bound to be prone to errors. As a result, the breakup behaviors on different grids are dissimilar. Increasing the grid resolution prolongs the instant in time where a subgrid structure will form during a hydrodynamic breakup event. Nevertheless, for most part of the retraction process, even the coarsest ($D/\Delta x = 8$) grid resolved the underlying physics adequately. A similar observation regarding minimum grid size requirement in the context of atomization was recently made by Shinjo and Umemura [101].

4. Conclusion

We have presented the results of a number of validation and verification exercises for the VoF open source solver *interFoam*, which forms part of the computational toolkit OpenFOAM. The solver represents a variant of the VoF methodology, which incorporates a compressive flux, rather than subgrid geometric reconstruction. For completeness, a full description of the algorithm, which includes the transport of the liquid fraction, the solution of momentum and the pressure iteration (PISO), is provided within a two-phase flow context. In each category of testing, our findings are as follows.

- (i) Tests of kinematics showed excellent mass conservation and acceptable advection errors. For the notched disc in rotational flow, the rate of convergence was found to be close to unity. However, for a deforming globule, it was slightly improved (~ 1.5). This is comparable to the existing methodologies [46], which have a similar degree of maturity. However, it is noticeably worse when compared with the latest algorithms which include subgrid level geometric reconstruction. In addition, a notable disparity is found between the local truncation and the global error.
- (ii) The solver was tested on inertia-dominated flows ($We \gg 1$) and was found to perform extremely well. Even with modest grid resolutions, accurate physics was captured for the two unsteady problems. Similar observations have been presented in the literature for flows in this high Weber number regime [3, 7, 17].
- (iii) In relation to surface tension-dominated flows, we performed several verification tests and showed that the momentum formulation in *interFoam* is indeed designed to yield a discrete balance between pressure gradient and surface tension. The performance of the solver is comparable with the face centered balanced force approach of Francois *et al* [1]. An interesting finding in this work is regarding the amplification of magnitude spurious currents with the factor $1/\rho$ (section 3.3.1). It is shown that this factor amplifies an imbalance that originates from residuals in the solution of the pressure equation and not from any algorithmic inconsistencies.
- (iv) Accuracy of curvature computation was also evaluated through a verification test involving a stationary 2D droplet. The finding is that the average curvature and the average pressure jump across the interface do not converge to the analytical prediction. Such a convergence to a slightly erroneous value has already been reported in the literature [32] for VoF methods. These curvature errors, however, did not create any noticeable issues, and good agreement with experiments was obtained.

- (v) Regarding spurious currents, following [66], we showed that for `interFoam`, the growth of spurious currents can be controlled by choosing a time step consistent with the following:

$$\Delta t \leq \max(10\tau_\mu, 0.1\tau_\rho) \quad \text{or, in the form of } \tau_\sigma,$$

$$\Delta t \leq \tau_\sigma = \frac{1}{2} \left\{ 10\tau_\mu + \sqrt{(10\tau_\mu)^2 + 4 \times 0.01\tau_\rho^2} \right\}. \quad (84)$$

- (vi) Finally, regarding computations that involved atomization, `interFoam` was able to capture the physics even with modest levels of grid resolution. This was exemplified in the test cases corresponding to a 2D standing capillary wave, Rayleigh breakup of a laminar jet and capillary retraction of a liquid jet.

Acknowledgments

Support from the Office of Naval Research through Mark Spector, code 331, and Caterpillar Corporation is acknowledged. The authors thank H Weller, H Jasak and the other members of OpenCFD for making OpenFOAM freely available for research. Our thanks are also due to members of our group, Steven Lewis and Eelco Gehring, for their assistance in completing tests related to droplet impingement (section 3.2.2).

References

- [1] Francois M M, Cummins S J, Dendy E D, Kothe D B, Sicilian J M and 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** 141–73
- [2] Shinjo J and Umemura A 2011 Detailed simulation of primary atomization mechanisms in diesel jet sprays (isolated identification of liquid jet tip effects) *Proc. Combust. Inst.* **33** 2089–97
- [3] Deshpande S S, Trujillo M F, Wu X and Chahine G L 2012 Computational and experimental characterization of a liquid jet plunging into a quiescent pool at shallow inclination *Int. J. Heat Fluid Flow* **34** 1–14
- [4] Prosperetti A and Tryggvason G 2009 *Computational Methods for Multiphase Flow* (Cambridge: Cambridge University Press)
- [5] Hirt C W and Nichols B D 1981 Volume of fluid method for the dynamics of free boundaries *J. Comput. Phys.* **39** 201–25
- [6] Osher S and Sethian J 1988 Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations *J. Comput. Phys.* **79** 12–49
- [7] Berberovic E, van Hinsberg N P, Jakirlic S, Roisman I V and Tropea C 2009 Drop impact onto a liquid layer of finite thickness: dynamics of the cavity evolution *Phys. Rev. E* **79** 036306-15
- [8] Maiwald A and Schwarze R 2011 Numerical analysis of flow-induced gas entrainment in roll coating *Appl. Math. Modelling* **35** 3516–26
- [9] Saito K, Srinivasan V and Salazar A J 2011 Modeling the disintegration of modulated liquid jets using volume-of-fluid (VoF) methodology *Appl. Math. Modelling* **35** 3710–30
- [10] Liu X and Garcia M H 2008 Three-dimensional numerical model with free water surface and mesh deformation for local sediment scour *J. Waterway Port Coastal Ocean Eng.* **134** 203–17
- [11] Ishimoto J, Sato F and Sato G 2010 Computational prediction of the effect of microcavitation on an atomization mechanism in a gasoline injector nozzle *J. Eng. Gas Turbines Power* **132** 082801-1–15
- [12] Raach H, Somasundaram S and Mitrovic J 2011 Optimisation of turbulence wire spacing in falling films performed with openoam *Desalination* **267** 118–19
- [13] Roisman I V, Weickgenannt C M, Lembach A N and Tropea C 2010 Drop impact close to a pore: experimental and numerical investigations *ILASS—Europe 2010, 23rd Annual Conf. on Liquid Atomization and Spray Systems (Brno, Czech Republic, September 2010)* 1–6
- [14] Saha A A and Mitra S K 2009 Effect of dynamic contact angle in a volume of fluid (VoF) model for a microfluidic capillary flow *J. Colloid Interface Sci.* **339** 461–80
- [15] Saha A A, Mitra S K, Tweedie M, Roy S and McLaughlin J 2009 Experimental and numerical investigation of capillary flow in SU8 and PDMS microchannels with integrated pillars *Microfluid Nanofluid* **7** 451–65
- [16] Gopala V R, Jan-Aiso Lycklama a, Nijeholt Haverkate B and Bakker P 2011 Development and validation of a CFD model predicting the backfill process of a nuclear waste gallery *Nucl. Eng. Des.* **241** 2508–18

- [17] Trujillo M F, Alvarado J, Gehring E and Soriano G S 2011 Numerical simulations and experimental characterization of heat transfer from a periodic impingement of droplets *J. Heat Transfer* **133** 1–10
- [18] Ubbink O 1997 Numerical prediction of two fluid systems with sharp interfaces *PhD Thesis* Imperial College of Science Technology and Medicine
- [19] Jasak H 2009 Openfoam: open source CFD in research and industry *Int. J. Nav. Archit. Ocean Eng.* **1** 89–94
- [20] Weller H G, Tabor G, Jasak H and Fureby C 1998 A tensorial approach to computational continuum mechanics using object-oriented techniques *Comput. Phys.* **12** 620–31
- [21] Bolton B and Middleman S 1980 Air entrainment in a roll coating system *Chem. Eng. Sci.* **35** 597–601
- [22] Chaves H, Obermeier F and Seidel T 2000 Fundamental investigation of disintegration of a sinusoidally forced liquid jet *8th Int. Conf. on Liquid Atomization and Spray Systems (Pasadena, USA)* pp 1018–25
- [23] Raach H and Somasundaram S 2008 Numerical investigations on heat transfer in falling films around turbulence wires *Proc. Int. Conf. 'Eurotherm 2008' (Eindhoven, the Netherlands)*
- [24] Zeng J 2007 On modeling of capillary filling http://www.coventor.com/pdfs/on_modeling_of_capillary_filling.pdf
- [25] Soriano G, Alvarado J and Lin Y 2010 Experimental characterization of single and multiple droplet impingement on surfaces subject to constant heat flux conditions *IHTC14—Proc. Int. Heat Transfer Conf.* 707–15
- [26] Sussman M, Smereka P and Osher S 1994 A level set approach for computing solutions to incompressible two phase flow *J. Comput. Phys.* **114** 146–59
- [27] Tryggvason G, Scardovelli R and Zaleski S 2011 *Direct Numerical Simulations of Gas–Liquid Multiphase Flows* (Cambridge: Cambridge University Press)
- [28] Brackbill J U, Kothe D B and Zemach C 1992 A continuum method for modeling surface tension *J. Comput. Phys.* **100** 335–54
- [29] Tryggvason G, Bunner B, Ebrat O and Tauber W 1998 Computations of multiphase flows by a finite difference/front tracking method. i Multi-fluid flows *Lecture Series 1998–03, 29th Computational Fluid Dynamics (Von Karman Institute for Fluid Dynamics, 1998)*
- [30] Rider W J and Kothe D B 1998 Reconstructing volume tracking *J. Comput. Phys.* **141** 112–52
- [31] Harvie D J E and Fletcher D F 2000 A new volume of fluid advection algorithm: the stream scheme *J. Comput. Phys.* **162** 1–32
- [32] Renardy Y and Renardy M 2002 Prost: a parabolic reconstruction of surface tension for the volume-of-fluid method *J. Comput. Phys.* **183** 400–21
- [33] Aulisa E, Manservigi S and Scardovelli R 2003 A mixed markers and volume-of-fluid method for the reconstruction and advection of interfaces in two-phase and free-boundary flows *J. Comput. Phys.* **188** 611–39
- [34] Hysing S 2006 A new implicit surface tension implementation for interfacial flows *Int. J. Numer. Methods Fluids* **51** 659–72
- [35] Williams M W, Kothe D B and Puckett E G 1998 *Accuracy and Convergence of Continuum Surface Tension Models Fluid dynamics at interfaces* ed W Shyy and R Narayanan (Cambridge: Cambridge University Press) pp 294–305
- [36] Desjardins O, Moureau V and Pitsch H 2008 An accurate conservative level set/ghost fluid method for simulating turbulent atomization *J. Comput. Phys.* **227** 8395–9416
- [37] Meier M, Yadigaroglu G and Smith B L 2002 A novel technique for including surface tension in PLIC-VoF methods *Eur. J. Mech. B* **21** 61–73
- [38] Scardovelli R and Zaleski S 2003 Interface reconstruction with least-square fit and split Eulerian–Lagrangian advection *Int. J. Numer. Methods Fluids* **41** 251–74
- [39] Jamet D, Torres D and Brackbill J U 2002 On the theory and computation of surface tension: the elimination of parasitic currents through energy conservation in the second-gradient method *J. Comput. Phys.* **182** 262–76
- [40] Lopez J and Hernandez J 2010 On reducing interface curvature computation errors in the height function technique *J. Comput. Phys.* **229** 4855–68
- [41] Desjardins O and Moureau V 2010 Methods for multiphase flows with high density ratio *Technical Report, Proc. of the Summer Program* Center for Turbulence Research
- [42] Umemura A 2011 Self-stabilizing mechanism of a laminar inviscid liquid jet issuing from a circular nozzle *Phys. Rev. E* **83** 046307-1–19
- [43] Raessi M and Pitsch H 2010 A level-set based methodology for modeling interfacial flows characterized by large density ratios *ILASS-Americas 22nd Annual Conf. on Liquid Atomization and Spray Systems (Cincinnati, OH, May 2010)*
- [44] Ghods S and Herrmann M 2011 A method for modeling multiphase flows characterized by large density ratios *ILASS-Americas 23rd Annual Conf. on Liquid Atomization and Spray Systems (Ventura, CA, May 2011)*
- [45] Gorokhovskii M and Herrmann M 2008 Modeling primary atomization *Annu. Rev. Fluid Mech.* **40** 343–66

- [46] Xiao F, Li S and Chen C 2011 Revisit to the thinc scheme: a simple algebraic VoF algorithm *J. Comput. Phys.* **230** 7086–92
- [47] Ubbink O and Issa R I 1999 A method for capturing sharp fluid interfaces on arbitrary meshes *J. Comput. Phys.* **153** 26–50
- [48] Enright D, Fedkiw R, Ferziger J and Mitchell I 2002 A hybrid particle level set method for improved interface capturing *J. Comput. Phys.* **183** 83–116
- [49] Menard T, Tanguy S and Berlemont A 2007 Coupling level set/VoF/ghost fluid methods: validation and application to 3d simulation of the primary break-up of a liquid jet *Int. J. Multiphase Flow* **33** 510–24
- [50] Zalesak S T 1979 Fully multidimensional flux-corrected transport algorithm for fluids *J. Comput. Phys.* **31** 335–62
- [51] Cummins S J, Francois M M and Kothe D B 2005 Estimating curvature from volume fractions *Comput. Struct.* **83** 425–34
- [52] Popinet S 2009 An accurate adaptive solver for surface-tension-driven interfacial flows *J. Comput. Phys.* **228** 5838–66
- [53] Lossaso F, Fedkiw R and Osher S 2006 Spatially adaptive techniques for level set methods and incompressible flow *Comput. Fluids* **35** 995–1010
- [54] Wang Z, Yang J and Stern F 2009 An improved particle correction procedure for the particle level set method *J. Comput. Phys.* **228** 5819–37
- [55] Sussman M and Puckett E G 2000 A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows *J. Comput. Phys.* **162** 301–37
- [56] Nave J C, Rosales R R and Seibold B 2010 A gradient-augmented level set method with an optimally local coherent advection scheme *J. Comput. Phys.* **229** 3802–27
- [57] Chopp D 1993 Computing minimal surfaces via level set curvature flow *J. Comput. Phys.* **106** 77–91
- [58] Sussman M and Fatemi E 1999 An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow *SIAM J. Sci. Comput.* **20** 1165–91
- [59] Russo G and Smereka P 2000 A remark on computing distance functions *J. Comput. Phys.* **163** 51–67
- [60] Anumolu L and Trujillo M F 2011 Redistancing the augmented level set method ILASS—Americas 23rd Annual Conf. on Liquid Atomization and Spray Systems (Ventura CA, May 2011)
- [61] Anumolu L and Trujillo M F 2012 A hybrid re-initialization scheme for the augmented level set method ILASS—Americas 24th Annual Conf. on Liquid Atomization and Spray Systems (San Antonio, TX, May 2012)
- [62] Rudman M 1997 Volume-tracking methods for interfacial flow calculations *Int. J. Numer. Methods Fluids* **24** 671–91
- [63] Popinet S and Zaleski S 1999 A front tracking algorithm for accurate representation of surface tension *Int. J. Numer. Methods Fluids* **30** 775–93
- [64] Shirani E, Ashgriz N and Mostaghimi J 2005 Interface pressure calculation based on conservation of momentum for front capturing methods *J. Comput. Phys.* **203** 154–75
- [65] Tanguy S, Menard A and Berlemont T 2007 A level set method for vaporizing two-phase flows *J. Comput. Phys.* **221** 837–53
- [66] Galusinski C and Vigneaux P 2008 On stability condition for bifluid flows with surface tension: application to microfluidics *J. Comput. Phys.* **227** 6140–64
- [67] Raessi M and Pitsch H 2009 Modeling interfacial flows characterized by large density ratios with the level set method *Technical Report* Center for Turbulence Research, Stanford University
- [68] Youngs D L 1982 Time dependent multi-material flow with large fluid distortions *Numerical Methods for Fluid Dynamics* ed K W Morton and M J Baines (New York: Academic) pp 273–85
- [69] Diwakar S V, Das S K and Sundararajan T 2009 A quadratic spline based interface (quasi) reconstruction algorithm for accurate tracking of two-phase flows *J. Comput. Phys.* **228** 9107–30
- [70] OpenFOAM User Guide 2008 <http://www.openfoam.org/docs/>
- [71] OpenFOAM-1.5 2008 *Openfoam: The Open Source CFD Toolbox* <http://www.openfoam.com/>
- [72] van Leer B 1974 Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second order scheme *J. Comput. Phys.* **14** 361–70
- [73] Roe P L 1986 Characteristic-based schemes for the euler equations *Annu. Rev. Fluid Mech.* **18** 337–65
- [74] Leonard B P 1979 A stable and accurate convective modeling procedure based on quadratic upstream interpolation *Comput. Methods Appl. Mech. Eng.* **19** 59–98
- [75] Jasak H, Weller H G and Gosman A D 1999 High resolution NVD differencing scheme for arbitrarily unstructured meshes *Int. J. Numer. Methods Fluids* **31** 431–49
- [76] Pozrikidis C 1997 *Introduction to Theoretical and Computational Fluid Dynamics* (Oxford: Oxford University Press)

- [77] Rusche H 2002 Computational fluid dynamics of dispersed two-phase flows at high phase fractions *PhD Thesis* Imperial College of Science Technology and Medicine
- [78] Issa R I 1986 Solution of the implicitly discretised fluid flow equations by operator splitting *J. Comput. Phys.* **62** 40–65
- [79] Gopala V R and van Wachem B G M 2008 Volume of fluid methods for immiscible-fluid and free-surface flows *Chem. Eng. J.* **141** 204–21
- [80] Pilliod J E Jr and Puckett E G 2004 Second-order accurate volume-of-fluid algorithms for tracking material interfaces *J. Comput. Phys.* **199** 465–502
- [81] Wang Z, Yang J, Koo F and Stern B 2009 A coupled level set and volume-of-fluid method for sharp interface simulation of plunging breaking waves *Int. J. Multiphase Flow* **35** 227–46
- [82] Wang Z, Yang J and Stern F 2012 A new volume-of-fluid method with a constructed distance function on general structured grids *J. Comput. Phys.* **231** 3703–22
- [83] Book D L and Boris J P 1973 Flux-corrected transport. I. Shasta, a fluid transport algorithm that works *J. Comput. Phys.* **11** 38–69
- [84] Jasak H and Weller H G 1995 Interface-tracking capabilities of the intergamma differencing scheme *Technical Report* Imperial College, University of London, London
- [85] Aulisa E, Manservigi S and Scardovelli R 2006 A novel representation of the surface tension force for two-phase flow with reduced spurious currents *Comput. Methods Appl. Mech. Eng.* **195** 6239–57
- [86] Hernandez J, Lopez J, Gomez P, Zanzi C and Faura F 2008 A new volume of fluid method in three dimensions. I: Multidimensional advection method with face-matched flux polyhedra *Int. J. Numer. Methods Fluids* **58** 897–921
- [87] Wang Y, Simakhina S and Sussman M 2012 A hybrid level set-volume constraint method for incompressible two-phase flow *J. Comput. Phys.* **231** 6438–71
- [88] Diskin B and Thomas J L 2010 Notes on accuracy of finite-volume discretization schemes on irregular grids *Appl. Numer. Math.* **60** 224–60
- [89] Diskin B and Thomas J L 2011 Notes on accuracy of finite-volume discretization schemes on irregular grids http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20110011294_2011011144.pdf
- [90] Syrakos A and Goulas A 2006 Estimate of the truncation error of finite volume discretization of the Navier–Stokes equations on colocated grids *Int. J. Numer. Methods Fluids* **50** 103–30
- [91] Kissling K, Springer J, Jasak H, Schutz S, Urban K and Piesche M 2010 A coupled pressure based solution algorithm based on the volume-of-fluid approach for two or more immiscible fluids *ECCOMAS CFD: 5th European Conf. on Computational Fluid Dynamics*
- [92] Lamb H 1932 *Hydrodynamics* (Cambridge: Cambridge University Press)
- [93] Sussman M 2003 A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles *J. Comput. Phys.* **187** 110–36
- [94] Cossali G E, Marengo M, Coghe A and Zhdanov S 2004 The role of time in single drop splash on thin film *Exp. Fluids* **36** 888–900
- [95] Cossali G E, Coghe A and Marengo M 1997 Impact of a single drop on a wetted solid surface *Exp. Fluids* **22** 463–72
- [96] Scardovelli R and Zaleski S 1999 Direct numerical simulation of free-surface and interfacial flow *Annu. Rev. Fluid Mech.* **31** 567–603
- [97] Harvie D J E, Davidson M R and Rudman M 2006 An analysis of parasitic current generation in Volume of Fluid simulations *Appl. Math. Modelling* **30** 1056–66
- [98] Rayleigh L 1879 On the instability of jets *Proc. Lond. Math. Soc.* **10** 4–13
- [99] Donnelly R J and Glaberson W 1966 Experiments on the capillary instability of a liquid jet *Proc. R. Soc. Lond. A* **290** 547–56
- [100] Lafrance P 1975 Nonlinear breakup of a laminar liquid jet *Phys. Fluids* **18** 428–32
- [101] Shinjo J and Umemura A 2010 Simulation of liquid jet primary breakup: dynamics of ligament and droplet formation *Int. J. Multiphase Flow* **36** 513–32