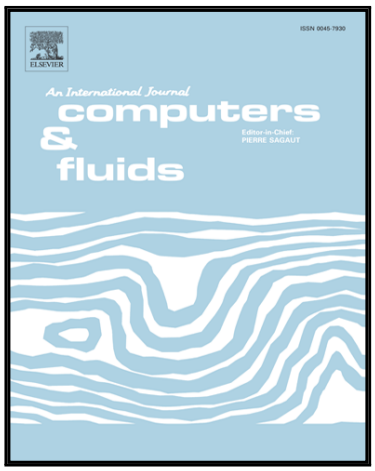


Simulation of micro-flow dynamics at low capillary numbers using adaptive interface compression

M. Aboukhedr, A. Georgoulas, M. Marengo, M. Gavaises, K. Vogiatzaki

PII: S0045-7930(18)30009-4
DOI: 10.1016/j.compfluid.2018.01.009
Reference: CAF 3692

To appear in: *Computers and Fluids*

Received date: 12 July 2017
Revised date: 15 November 2017
Accepted date: 13 January 2018

Please cite this article as: M. Aboukhedr, A. Georgoulas, M. Marengo, M. Gavaises, K. Vogiatzaki, Simulation of micro-flow dynamics at low capillary numbers using adaptive interface compression, *Computers and Fluids* (2018), doi: 10.1016/j.compfluid.2018.01.009

Highlights

- Multiphase flow solver using adaptive compression scheme has been introduced.
- Wide range of conditions using well-established benchmark cases has been tested.
- The adaptive compression facilitates simulating flows at law capillary numbers.
- The adaptive nature of the coef. counter balances the need for very fine grids.
- Using the mentioned method gives accurate results in estimating bubble formation.

Simulation of micro-flow dynamics at low capillary numbers using adaptive interface compression ☆

M. Aboukhedr[a,*], A. Georgoulas[b], M. Marengo[b], M. Gavaises[a], K. Vogiatzaki[b]

[a]*Department of Mechanical Engineering, City, University of London, UK*
[b]*School of Computing, Engineering and Mathematics, Advanced Engineering Centre, University of Brighton, Brighton, UK*

Abstract

A numerical framework for modelling micro-scale multiphase flows with sharp interfaces has been developed. The suggested methodology is targeting the efficient and yet rigorous simulation of complex interface motion at capillary dominated flows (low capillary number). Such flows are encountered in various configurations ranging from micro-devices to naturally occurring porous media. The methodology uses as a basis the Volume-of-Fluid (VoF) method combined with additional sharpening smoothing and filtering algorithms for the interface capturing. These algorithms help the minimisation of the parasitic currents present in flow simulations, when viscous forces and surface tension dominate inertial forces, like in porous media. The framework is implemented within a finite volume code (OpenFOAM) using a limited Multi-dimensional Universal Limiter with Explicit Solution (MULES) implicit formulation, which allows larger time steps at low capillary numbers to be utilised. In addition, an adaptive interface compression scheme is introduced for the first time in order to allow for a dynamic estimation of the compressive velocity only at the areas of interest and thus has the advantage of avoiding the use of a-priori defined parameters. The adaptive method is found to increase the numerical accuracy and to reduce the sensitivity of the methodology to tuning parameters. The accuracy and stability of the proposed model is verified against five different benchmark test cases. Moreover, numerical results are compared against analytical solutions as well as available experimental data, which reveal improved solutions relative to the standard VoF solver.

Keywords: CFD, interFoam, two-phase flows, microfluidics, surface tension forces, parasitic currents, micro-scale modelling

☆This document is a collaborative effort

*Corresponding author
Email address: mahmoud.aboukhedr.1@city.ac.uk (M. Aboukhedr)

List of Nomenclature

u	Velocity
p	Pressure
p_c	Capillary pressure
p_d	Dynamic pressure
f	External forces
f_g	Gravitational forces
f_s	Surface tension force
ρ	Density
μ	Dynamic viscosity
$u_{r,f}$	Relative velocity at cell faces
σ	Surface tension
ϕ_f	Volumetric flux
ϕ_c	Compression volumetric flux
ϕ	Capillary flux
$\phi_{threshold}$	Threshold volumetric flux
V_i	Volume per grid cell
S_f	Outward-pointing face area
κ	Interface curvature
κ_f	Filtered interface curvature calculated based on smooth function α_{smooth}
$\kappa_{s,i+1}$	Smooth interface curvature calculated based on smooth function κ_f
κ_{final}	Weighted interface curvature calculated based on smooth function $\kappa_{s,i}$
η_s	Normal vector to the interface
δ_s	Dirac delta function
α	Volume fraction
α_{smooth}	Volume fraction using Laplacian formulation
α_{sh}	Sharp inductor function
$C_{compr.}$	Constant interface compression coefficient
C_{adp}	Adaptive interface compression
C_{sh}	Sharpening coefficient
U_f	filtering coefficient
$\langle \eta_s \rangle_f$	Face centred normal vector
$\langle \nabla \alpha \rangle_f$	Volume fraction interpolated from cell centre to face centre
δ_n	Small value

1. Introduction

Flows through "narrow passages" such as micro-channels or pore-scale flows whose dimensions are less than O(mm) and greater than O(μm) differ from their macroscopic counterparts at important aspects: the small size of the geometries makes molecular effects such as wall slip or wettability more important,

while amplifies the magnitudes of certain ordinary continuum effects associated with strain rate and shear stress. Such flows are present in various natural formations (rocks and human organs) as well as man-made applications (micro-conductors, micro-emulsions, etc.). Thus, microscale physics attracts the interest of various disciplines including cosmetic and pharmaceutical industries as well as biomedical and petroleum engineering. For more details on the application of microscale geometries, the reader is referred to [1]. Among all these applications transportation of droplets in microchannels at low Capillary ($Ca = \frac{\mu u}{\sigma}$) numbers has attracted the interest of researchers from the theoretical and experimental point of view [2, 3, 4]. For example, understanding the dynamics of immiscible fluids in micro-devices can facilitate the creation of monodisperse emulsions. Droplets of the same size move with low velocities through microchannel networks and are used as micro-reactors to study very fast chemical kinetics [5]. Another example of low Ca flow dynamics in micro-scale can be seen at trapped oil blobs in porous reservoirs. Understanding the trapping flow dynamics at the pore scale level can be the key to minimising the trapping of a non-wetting phase and enhancing recovery systems of hydrocarbons, [6]. Although a large number of methods has been developed for simulating multiphase flows at macro-scale including the well known Level Sets (LS) [7] and Volume of Fluid (VoF) methods [8], the extension of these methods to micro-scale is not always straightforward. The main weakness of the LS methods is that they do not preserve mass. As a result, poorly resolved regions of the flow are typically susceptible to mass loss behaviour and loss of signed distance property due to advection errors. Various modification have been suggested focusing on solving the conservation issues [9], extending the method to high Reynolds numbers [10] and to unstructured meshes [11, 12]. While using a re-initialization procedure as discussed by [13] is a solution to the mass conservation issue, it increases the computational cost and creates an artificial interface displacement that may affect mass conservation, see the review by Russo and Smereka [14] for details. Similarly the VoF method is based on the numerical solution of a transport equation that distinguishes the two fluids in the domain, and it represents the volume percentage of each fluid phase in each cell over the total volume of the cell. The interface between the two phases is defined in the cells where the VoF function takes a value between (0, 1). In incompressible flows, the mass conservation is achieved by using either a geometrical reconstruction coupled with a geometrical approximation of the volume of fluid advection or a compressive scheme as discussed by Rusche [15] and implemented by Weller et al. [16]. The VoF method has been the most widely used interface capturing

method due to ease of implementation as reviewed by Wörner [2].

Within the VoF framework two commonly used methods for interface representation exist: (a) a compressive method and (b) a geometric method. Both VoF methods are used in order to calculate the discrete volume fraction of each phase within a cell, which is then transported based on the underlying fluid velocity. Compressive VoF methods discretise the partial differential equation describing the transport of the volume fraction of each phase using algebraic differencing schemes [17, 18]. The key for the accuracy of these methods is that, in order to keep the interface sharp and without distortion, the temporal and spatial discretisation should be performed using higher order schemes and careful tuning. Otherwise the method may suffer from excessive diffusion of the interface region which also affects the calculation of the interface curvature and the normal interface vectors. Park et al. [19] and Gopala and van Wachem [20] showed the compressive VoF methods capabilities of advecting sharp interface, and they also underlined the difficulties in retaining the shape and sharpness of the interface. Using a geometric method, an explicit representation of the interface is advected, reconstructed from the VoF volume fraction field. The piecewise linear methods so-called (PLIC) is the most developed reconstruction method found in the literature [21, 22]. Geometric methods advect the interface very accurately, but their main drawback is their complexity for 3D applications, in particular when used in conjunction with an unstructured mesh [23].

Recently, the coupling between VoF and LS, the so-called Coupled Level Set Volume Of Fluid (CLSVoF) method [24] has also received significant attention since it combines the advantages of both methods, i.e., the VoF mass conservation and the LS interface sharpness [24, 25] . On the downside, this approach also combines the weaknesses of each method since techniques to keep the VoF interface sharp and reinitialise the distancing function are needed. Based on various published results for both methods [20, 26, 27, 28] the existent frameworks reviewed in the previous paragraph - regardless of the various modifications available - still suffer from their inherent severe drawbacks. These drawbacks are more pronounced in low Ca flows, and, as discussed in detail in Popinet and Zaleski [29], Tryggvason et al. [30] and Bilger et al. [31], stem from the fact that sharp discontinuities such as interfaces are represented by finite volume integrals [8]. The most common issue is that in all implicit interface capturing methods, the interface location is known by defining the normal and the curvature implicitly. For the VoF methods ,in particular, which are based on the representation of the discontinuous interface with continuous colour function, the calculation of the proper-

ties of each phase is possible, given an accurate numerical scheme for solving the colour function transport equation is available. However, the accuracy of the calculated interface curvature (that is then required for the calculation of the capillary pressure force) depends on determining the derivative of the introduced discontinuous colour function, which is considered to be difficult from a numerical point of view, and may leads to numerical instabilities [32].

An additional issue is the generation of non-physical velocities at the interface which are known as ”spurious” or ”parasitic” currents. The primary sources of spurious currents have been identified as the combination of inaccurate interface curvature and lack of a discrete force balance as discussed by Francois et al. [33]. It should be stressed that the local force imbalance between the capillary pressure and the pressure arising from the normal component of the surface tension force vectors (due to the imprecise evaluation of the local curvature) can create the non-physical velocities, (spurious currents”) which are commonly small in absolute values in inertia dominated flows, but become very problematic in capillary dominated flows.

Numerical challenges related to the advection of the interface in the context of VoF are well documented by Tryggvason et al. [30]. Intrinsic to the method, regardless if geometric reconstruction or interface compression is used, is the numerical diffusion of the interface, which is highly dependent on the mesh size [18]. The numerical diffusion can be reduced by using a geometrical reconstruction coupled with a geometrical approximation of the VoF advection as discussed by Roenby et al. [34]. Alternatively, using a compressive algorithm, the convective term of the VoF equation can be discretised using a compressive differencing scheme designed to preserve the interface sharpness. Examples include the HRIC by Muzaferija and Peric [35], or the compressive model available within OpenFoam [16]. Compression schemes do not require any geometrical reconstruction of the interface and extension to three dimensions and unstructured meshes is straightforward. However, compression schemes are not always sufficient to eliminate numerical diffusion completely and additional treatment is needed [36].

Various remedies that still have room for development have been suggested, and they can be summarised as following: (i) ensuring an accurate balance between local pressure and surface tension gradient. In Francois et al. [33] a cell-centered framework has been introduced. It is demonstrated that this algorithm can achieve an exact balance of between local pressure and surface tension gradient using structured mesh. Moreover, Francois et al. [33] and [37] discussed the origin of spurious currents within the introduced

balanced-force flow algorithms, as they highlighted the deficiencies introduced at the interface curvature estimation. (ii) sharp representation of the interface, with accurate curvature estimation and introduction of a so-called ”compression velocity” to damp diffusion. Ubbink and Issa [18] introduced the compressive discretisation scheme so-called Compressive Interface Capturing Scheme for Arbitrary Meshes CICSAM that makes a use of the normalised variable diagram concept introduced by Leonard [38]. Popinet [39] generalised a height-function and CSF formulations to an adaptive quad/octree discretisation to allow refinement along the interface for the case of capillary breakup of a three-dimensional liquid jet. Moreover, [39] discusses the long-standing problem of ”parasitic currents” around a stationary droplet in contrast to the recent study of Francois et al. [33], where the issue is shown to be solved by the combination of appropriate implementations of a balanced-force CSF approach and height-function curvature estimation. (iii) implicit or semi-implicit treatment of surface tension, Denner and van Wachem [40] reviewed the time-step requirements associated with resolving the dynamics of the equations governing capillary waves, to determine whether explicit and implicit treatments of surface tension have different time-step requirements with respect to the (1) dispersion of capillary waves, and (2) the formulation of an accurate time-step criterion for the propagation of capillary waves based on established numerical principles. The fully-coupled numerical framework with implicit coupling of the governing equations and the interface advection, and an implicit treatment of surface tension proposed by [40] was used to study the temporal resolution of capillary waves with explicit and implicit treatment of surface tension.

In the present work, a new framework for modelling immiscible two-phase flows for low Ca applications dominated by surface tension is suggested. The standard multiphase flow solver of OpenFOAM 2.3x has been extended to include sharpening and smoothing interface capturing techniques suitable for low Ca numbers flow. In addition a new generalised methodology that utilises an adaptive interface compression is introduced for the first time. While existing compression schemes are based on an a priori tuned parameter, which is typically kept constant throughout the simulations, in the present study compression is activated only in areas that the interface is prone to diffusion and the parameter is thus defined adaptively. This adaptive scheme is proved to limit the interface diffusion and to keep parasitic currents to minimal levels while reducing the computational time. The proposed framework for interface advection aspires to offer better modelling of flows in microscale that up to date have been proven problematic. The paper is structured

as following: Initially the numerical framework underlining the modifications suggested over the traditional VoF methodology in order to achieve better representation of the interface is introduced. The effect of each parameter used in the proposed framework is then evaluated individually based on a wide range of benchmark cases. The first test case refers to single and multiple droplet relaxations in a zero velocity field, aiming to assess the capability of the framework to damp spurious currents using various combination of control parameter. The evaluation of the solver for an advection test using the Zalesak disk [41] is also presented followed by results relevant to the motion of circle in a vortex field (Roenby et al. [34], Rider and Kothe [42]). Finally, a numerical study of the generation of bubbles in a T-junction is studied to evaluate the introduced framework in simulating more complex two-phase flows at a low Ca numbers.

2. Numerical method

The method presented in this section is implemented within the open source CFD toolkit OpenFOAM [43]. An incompressible and isothermal two-phase flow with constant phase densities ρ_1 and ρ_2 and viscosities μ_1 and μ_2 is considered. The two phases are treated as one fluid and a single set of equations is solved in the entire computational domain. The volume fraction, α of each phase within a cell is defined by an additional transport equation. The formulation for the conservation of mass and momentum for the phase mixture is given by the following equations:

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\frac{D}{Dt}(\rho \mathbf{u}) = \nabla \cdot \mathbf{T} - \nabla p + \boldsymbol{f} \tag{2}$$

where $\mathbf{u}$ is the fluid velocity, p is the pressure and ρ is the density. The pressure-velocity coupling is handled using the Pressure-Implicit with Splitting Operators(PISO) method of [44, 45]. The term $\nabla \cdot \mathbf{T} = \nabla \cdot (\mu \nabla \mathbf{u}) + \nabla \mathbf{u} \cdot \nabla \mu$ is the viscous stress tensor. The term $\boldsymbol{f} = \boldsymbol{f_g} + \boldsymbol{f_s}$ corresponds to all the external forces, i.e. $\boldsymbol{f_g} = \rho \boldsymbol{g}$ is the gravitational force and $\boldsymbol{f_s}$ represents the capillary forces for the case of constant surface tension coefficient σ. The global properties are weighted averages of the phase properties through the volume fraction value that is calculated in each cell:

$$\rho = \rho_1 + (\rho_2 - \rho_1)\alpha \tag{3}$$

$$\mu = \mu_1 + (\mu_2 - \mu_1)\alpha \tag{4}$$

The sharp interface Γ represents a discontinuous change of the properties of the two fluids. The surface tension force must balance the jump in the stress tensor along the fluid interface. At each time step, the dynamics of the interface are determined by the Young-Laplace balance condition as;

$$\Delta P_{exact} = \sigma\kappa \tag{5}$$

accounting for a constant surface tension coefficient σ along the interface. The term κ represents the interface curvature. The term on the right-hand side of Eq. 5 is effectively the source term in the Navier–Stokes equations for the singular capillary force, that is only present at the interface. In the proposed numerical method, the Continuum Surface Force (CSF) description of Brackbill et al. [8] is used to represent the surface tension forces in the following form:

$$f_s = \sigma\kappa_{final}\delta_s \tag{6}$$

where the term κ_{final} represents the interface curvature at the final stage of smoothing as discussed in section 2.2, δ_s is a delta function defined on the interface, and η_s is the normal vector to the interface α_{smooth} as discussed in section 2.2 and is calculated by the following equation:

$$\eta_s = \frac{\nabla\alpha_{smooth}}{|\nabla\alpha_{smooth}|} \tag{7}$$

The terms δ_s and κ_f are associated with the artificially smoothed and sharpened indicator function fields that will be discussed in details in the following section. In the VoF method, the indicator function α represents the volume fraction of one of the fluid phases in each computational cell. The indicator function evolves spatially and temporally according to an advection transport equation of the following general form:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{u}) = 0 \tag{8}$$

Ideally, the interface between the two phases should be massless since it represents a sharp discontinuity. However, within VoF formulation the numerical diffusion of Eq. 8 results in values of α that vary between 0 and 1.

The framework described above reflects the generalised framework of VoF methods that has been used in an extensive range of two-phase flow problems with various adjustments and different degrees of success. In the following sub-sections, an enhanced version of this basic framework is presented; its validity is demonstrated through a range of benchmark cases that addresses some numerically challenging problems reported in the relevant literature.

2.1. Adaptive Compression Scheme (Implicit)

To deal with the problem of numerical diffusion of α, an extra compression term is used in order to limit the convection term of Eq. 8 and consequently the thickness of the interface. Its numerical significance relays on defining local flow (u) at the interface and preventing the increase of the gradient when alpha is not constant, (i.e. the absolute value of the time derivative increases to counterbalance). The model for the compression term makes use of the two-fluid Eulerian approach, where phase fraction equations are solved separately for each individual phase, assuming that the contributions of two fluids velocities for the free surface are proportional to the corresponding phase fraction. These phase velocities (u_1 and u_2) relate with the global velocity of the one fluid approach u as:

$$u = \alpha u_1 + (1 - \alpha) u_2 \tag{9}$$

Replacing the above equation to Eq. 8 one gets:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot \left\{ \left(\alpha u_1 + (1 - \alpha) u_2 \right) \alpha \right\} = 0 \tag{10}$$

Considering a relative velocity between the two phases (u_r=u_1-u_2) which arises from the density and viscosity stresses changes across the interface, the above equation can be written in terms of the velocity of

the fluid:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (u_1 \alpha) \underbrace{- \nabla \cdot \left\{ u_{r,f} \alpha \Big((1-\alpha) \Big) \right\}}_{\text{compression term}} = 0 \tag{11}$$

It should be noticed that in the above equation in the calculation of $\nabla \cdot (u\alpha)$ term the unknown velocity u_1 appears instead of u creating an inconsistency with the basic concept of the one fluid approach. However, since the compression term in reality is active only at the interface, continuity imposes $u_1 = u_2 = u$ and thus u_1 by u can be replaced. The discretisation of the compression term in Eq. 11 is not based directly on the calculation of the relative velocity u_r at cell faces from Eq. 9 since u_1 and u_2 are unknown. It is instead formulated based on the maximum velocity magnitude at the interface region and its direction, which is determined from the gradient of the phase fraction:

$$u_{r,f} = min\left(C_{compr.} \frac{|\phi_f|}{|S_f|}, max\left[\frac{|\phi_f|}{|S_f|} \right] \right) \left(\langle \eta_s \rangle_f \right) \tag{12}$$

where the term ϕ_f is the volumetric flux and S_f is the outward-pointing face area vector and $\langle \eta_s \rangle_f$ is the face centred interface normal vector. $\langle \rangle_f$ is used to denote interpolation from cell centres to face centres using a linear interpolation scheme, and defined as following:

$$\langle \eta_s \rangle_f = \frac{\langle \nabla \alpha \rangle_f}{|\langle \nabla \alpha \rangle_f + \delta_n|} \cdot S_f \tag{13}$$

and

$$\delta_n = \frac{1e^{-8}}{\left(\frac{\sum_N V_i}{N} \right)^{1/3}} \tag{14}$$

where δ_n is a small number to ensure that the denominator never becomes zero, N is the number of computational cells, for each grid block i and V_i is its volume

The compressive term is taken into consideration only at the interface region and it is calculated in the normal direction to the interface. The maximum operation in Eq. 12 is performed over the entire domain, while the minimum operation is done locally on each face. The constant ($C_{compr.}$) is a user-specified value, which serves as a tuning parameter. Depending on its value, different levels of compression result are calculated. For example, there is no compression for C= 0 while there is moderate compression with

C≤1 and enhanced compression for C≥1. In most of the simulations presented here ($C_{compr.}$) is taken as unity, after initial trial simulations. Values higher than unity in this case may lead to non-physical results. Generally, this compression factor can take values from 0 (no compression) up to 4 (maximum compression) as suggested in the literature; the selected values are case specific. To overcome the need for a priori tuning, in the present numerical framework a new adaptive algorithm has been implemented that is based on the idea of introducing instead of a constant value for $C_{compr.}$ a dynamic one C_{adp} through the following relation:

$$C_{adp} = \left| -\frac{u_n \cdot \nabla\alpha}{|u_n||\nabla\alpha|} \right| \tag{15}$$

$$\phi_c = max\left(C_{adp}, C_{compr.}\right)\frac{|\phi_f|}{|S_f|} \tag{16}$$

where ϕ_c is the compression volumetric flux calculated, u_n represents each phase velocity normal to the interface velocity. It is expressed as

$$u_n = (U \cdot n_s)x(n_s)x|\alpha - 0.01| * |0.99 - \alpha| \tag{17}$$

The concept of using u_n is shown in Fig. 1: when the interface profile becomes diffusive (wide) C_{adp} value will increase accordingly in the zone of interest, while when the profile is already sharp and additional compression is not necessary C_{adp} will go to zero. Note that the compression term in Eq. 11 is only valid for the cells at the interface. However, to solve Eq. 15, a wider region of α is required. Therefore, the facial cell field is extrapolated to a wider region using the expression (near interface) in Eq. 17 as ($|\alpha - 0.01| * |0.99 - \alpha|$). The new calculated, adaptive compression coefficient ϕ_c then substitutes the original $C_{compr.}\frac{|\phi_f|}{|S_f|}$ and Eq. 12 can be rewritten as:

$$u_{r,f} = min\left(\phi_c, max\left[\frac{|\phi_f|}{|S_f|}\right]\right)\left(\langle\eta_s\rangle_f\right) \tag{18}$$

The new equation still has a user defined value $C_{compr.}$ in cases when the adaptive coefficient is not sufficient.

2.2. Smoothing Scheme (Explicit)

By solving the transport equation for the volume fraction (Eq. 11), the value of (α) at the cell is updated. In order to proceed with the calculation of the interface surface scalar fields for the calculation of η_s and κ,

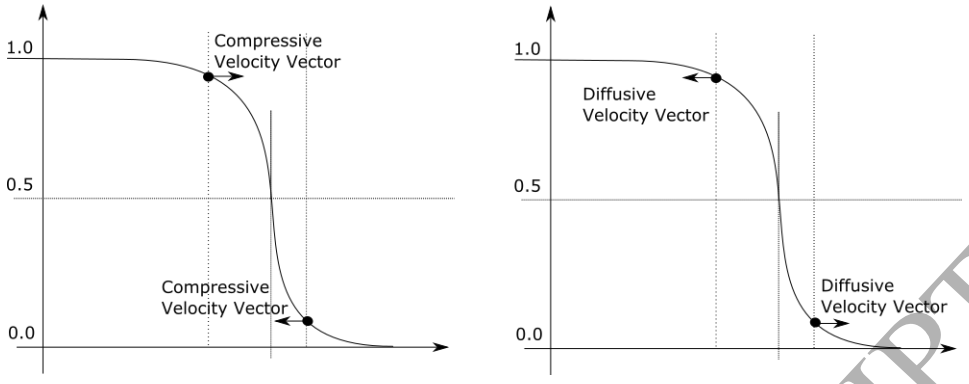


Figure 1: Schematic to represent the adaptive compression C_{adp} selection criteria

linear extrapolation from the cell centres is used. At this stage, the value of α sharply changes over a thin region as a result of the compression step. This abrupt change of the indicator function creates errors in calculating the normal vectors and the curvature of radius of the interface, which will be used to evaluate the interfacial forces. These errors induce non-physical parasitic currents in the interfacial region. A commonly followed approach in the literature to suppress these artefacts is to compute the interface curvature from a smoothed function α_{smooth}, which is calculated by the smoother proposed by Lafaurie et al. [17] and applied in OpenFOAM by Georgoulas et al. [46] and Raeini et al. [47]. The indicator function is artificially smoothed by interpolating it from cell centres to face centres and then back to the cell centres recursively using the following equation:

$$\alpha_{i+1} = 0.5\langle(\alpha_i)_{c\to f}\rangle_{f\to c} - 0.5\alpha_i \tag{19}$$

Initial trial simulations indicated that the recursive interpolation between the cell and face centres can be repeated up to three times, in order to prevent decoupling of the indicator function from the smoothed function. After smoothing is implemented, the interface normal vectors in the cells in the vicinity of the interface, are filtered using a Laplacian formulation. Equation 20 in Georgoulas et al. [46] is used in order to transform the VOF function (α_{i+1}) to a smoother function (α_{smooth}):

$$\alpha_{smooth} = \frac{\sum_{f=1}^{n}(\alpha_{i+1})_f S_f}{\sum_{f=1}^{n} S_f} \tag{20}$$

where the subscript denotes the face index (f) and (n) the times that the procedure is repeated in order

to get a smoothed field. The value at the face centre is calculated using linear interpolation. It should be stressed that smoothing tends to level out high curvature regions and should therefore be applied only up to the level that is strictly necessary to sufficiently suppress parasitic currents. After calculating the (α_{smooth}), the interface normal vectors are computed using 7, and the interface curvature at the cell centres can be obtained by $\kappa_f = -\nabla \cdot (\eta_s)$. Then in order to model the motion of the interfaces more accurately, an additional smoothing operation is performed to the curvature. The interface curvature in the direction normal to the interface is calculated, recursively for two iterations:

$$\kappa_{s,i+1} = 2\sqrt{\alpha_{smooth}(1-\alpha_{smooth})}\kappa_f + (1 - 2\sqrt{\alpha_{smooth}(1-\alpha_{smooth})}) * \frac{\left\langle \langle \kappa_{s,i}\sqrt{\alpha_{smooth}(1-\alpha_{smooth})}\rangle_{c\to f}\right\rangle_{f\to c}}{\left\langle \langle \sqrt{\alpha_{smooth}(1-\alpha_{smooth})}\rangle_{c\to f}\right\rangle_{f\to c}} \tag{21}$$

This additional smoothing procedure diffuses the variable κ_f away from the interface. Finally, the interface curvature at the face centres κ_{final} is calculated using a weighted interpolation method that is suggested by Renardy and Renardy [37]:

$$\kappa_{final} = \frac{\langle \kappa_{s,i}\sqrt{\alpha_{smooth}(1-\alpha_{smooth})}\rangle}{\langle \sqrt{\alpha_{smooth}(1-\alpha_{smooth})}\rangle} \tag{22}$$

where the interface curvature κ_{final} is obtained at face centres.

2.3. Sharpening Scheme (Explicit)

Recalling Eq. 6, the surface tension forces are calculated at the face centres based on the following equation:

$$f_s = (\sigma\kappa\delta_s)_f\dot{\eta}_s = \sigma\kappa_{final}\delta_{sf} \tag{23}$$

In order to control the sharpness of the surface tension forces, the delta δ_s is calculated from a sharpened indicator function α_{sh} as $\delta_s = \nabla_f^{\perp}\alpha_{sh}$, where $\nabla_f^{\perp}$ denotes the gradient normal to the face f. In Eq. 23 the surface tension force term is non-zero only at the faces across which the indicator function α_{sh} has values. The α_{sh} represents a modified indicator function, which is obtained by curtailing the original indicator function α as follows;

$$\alpha_{sh} = \frac{1}{1 - C_{sh}}\left[min\left(max(\alpha, 1 - \frac{C_{sh}}{2}), 1 - \frac{C_{sh}}{2}\right) - \frac{C_{sh}}{2}\right] \tag{24}$$

where C_{sh} is the sharpening coefficient. From Eq. 24 one can notice that, as the sharpening coefficient (C_{sh}) value increases, the unphysical interface diffusion decreases (i.e., it limits the effect of unphysical values at the interface, by imposing a restriction on alpha -α- as demonstrated). A zero value of C_{sh} will lead to the original CSF formulation, while as C_{sh} value increases the interface becomes sharper. As expected, the continuous -α_{smooth}- approach has a smooth (and diffused) transition across the interface, whereas the sharp $-\alpha_{sh}-$ approach has a more abrupt transition with larger extremes. At high values of C_{sh} (0.5 to 0.9), Eq. 24 limits the indicator function -α- where values between (0 to 0.4) are summed to zero and values between (0.6 to 1) are summed to be one. This implementation introduces a sharper approach of the surface tension forces as discussed by Aboukhedr et al. [48]. Values in the range of (0.5) C_{sh} were observed to give the best results for the most of our test cases.

2.4. Capillary Pressure Jump Modelling

In order to avoid difficulties associated with the discretisation of the capillary force f_c, rearrangement of the terms on the right hand side of the momentum equation is conducted following the work of [47], where Eq. 2 is rewritten in terms of the microscopic capillary pressure p_c:

$$\frac{D}{Dt}(\rho u) - \nabla \cdot T = -\nabla p_d + f', \tag{25}$$

$$f' = \rho g + f_s - \nabla p_c \tag{26}$$

where the dynamic pressure is defined as $p_d = p - p_c$. This approach includes explicitly the effect of capillary forces in the Navier-Stokes equations and allows for the filtering of the numerical errors related to the inaccurate calculation of capillary forces. Considering a static fluid configuration for a two phase flow, the stress tensor reduces to the form ($n \cdot \tau \cdot n = -p$), and the normal stress balance is assumed to have the form of ($p_c = \sigma \nabla \cdot n$) [49]. Then, the pressure jump across the interface is balanced by the curvature force at the interface.

$$\nabla \cdot \nabla p_c = \nabla \cdot f_s \tag{27}$$

Assuming that pressure jumps can sustain normal stress jumps across a fluid interface, they do not contribute to the tangential stress jump. Consequently, tangential surface stresses can only be balanced by viscous stresses. Therefore one can apply a boundary condition of:

$$\frac{\delta p_c}{\delta n_s} = 0 \tag{28}$$

where n_s is the normal direction to the boundaries. By including this set of equation to the Navier-Stokes equations, one can have a better balancing of momentum, hence filtering the numerical errors related to inaccurate calculations of the surface tension forces.

2.5. *Filtering numerical errors*

As the result of the numerical unbalance discussed in the previous sections when modelling the movement of a closed interface, it is difficult to maintain the zero-net capillary force, while modelling the movement of the interface. Hence it is difficult to decrease the errors in the calculation of capillary forces to zero $\oint f_s \cdot A_s = 0$ where A_s is the interface vector area. Raeini et al. [47] proposed as a solution to filter the non-physical fluxes generated due to the inconsistent calculation of capillary forces based on a user defined cut-off. The cut-off uses a thresholding scheme, aiming to filter the capillary fluxes ($\phi = |S_f|(f_s - \nabla_f^{\perp} p_c)$) and eliminate the problems related to the violation of the zero-net capillary force constraint on a closed interface. The proposed filtering procedure explicitly sets the capillary fluxes to zero when their magnitude is of the order of the numerical errors. The filter starts from setting an error threshold as:

$$\phi_{threshold} = U_f |f_s|_{avg} |S_f| \tag{29}$$

where $\phi_{threshold}$ is the threshold value below which capillary fluxes are set to zero and $|f|_{avg}$ is the average value of capillary forces over all faces. The filtering coefficient U_f is used to eliminate the errors in the capillary fluxes. Here a different U_f is used, so for different cases the U_f value will be set, which implies that the capillary fluxes are set to zero. After selecting the threshold, the capillary flux is filtered as:

$$\phi_{filter} = |S_f|(f - \nabla_f^{\perp} p_c) - max(min(|S_f|(f - \nabla_f^{\perp} p_c), \phi_{threshold}), -\phi_{threshold}) \tag{30}$$

Using this filtering method, numerical errors in capillary forces causing instabilities or introducing large errors in the velocity field are prevented. By using the aforementioned filtering technique, the problem of stiffness is found to be reduced by eliminating the high frequency capillary waves when the capillary forces are close to equilibrium with capillary pressure. Consequently, it allows larger time-steps to be used when modelling interface motion at low capillary numbers

3. Algorithm Implementation

The modelling approach for compression has been implemented using the OpenFOAM- Plus finite volume library [16], which is based on the VoF-based solver interFoam [50]. No geometric interface reconstruction or tracking is performed in interFoam; rather, a compressive velocity field is superimposed in the vicinity of the interface to counteract numerical diffusion as already discussed in section 2.1. In the original VoF-based solver (interFoam), the time step is only adjusted to satisfy the Courant-Friedrichs-Lewy (CFL) condition. A semi-implicit variant of MULES developed by OpenFOAM is used here which combines operator splitting with application of the MULES limiter to an explicit correction. It first executes an implicit predictor step, based on purely bounded numerical operators, before constructing an explicit correction on which the MULES limiter is applied. This approach maintains boundedness and stability at an arbitrarily large Courant number. Accuracy considerations generally dictate that the correction is updated and applied frequently, but the semi-implicit approach is overall substantially faster than the explicit method with its very strict limit on time-step. The indicator function is advected using Crank-Nicholson schemefor half of the time step using the fluxes at the beginning of each time step. Then the equations for the advection of the indicator function for the second half of the time step are solved iteratively in two loops. The discretised phase fraction (Eq. 11) is then solved for a user-defined number of sub-cycles (typically 2 to 3) using the multidimensional universal limiter with the [MULES] solver. Once the updated phase field is obtained, the algorithm enters in the pressure-velocity correction loop.

4. Results, Validation and Discussion

In the following sections, numerical simulations are presented for a range of benchmark cases that assess the performance of the proposed model. As a first benchmark case, a stationary single droplet and a pair of droplets (in the absence of gravity) have been considered. The convergence of velocity and capillary pressure to the theoretical solution is demonstrated. This test case assesses the performance of solvers in terms of spurious currents suppression. Then two other cases, commonly used in the literature, namely the Notched disc in rotating flow Zalesak [51] and the Circle in a vortex field Roenby et al. [34], Rider and Kothe [42] are examined. Finally, a more indicative example of flows through narrow passages is considered. This includes the generation of millimetric size bubbles in a T-junction. For the T-junction case, the prediction of any non-smoothed and diffused interface is accompanied by the development of spurious velocities resulting in unphysical results in comparison with the available experimental data. Calculations with the standard VoF-based solver of OpenFOAM (interFoam) are also included for completeness.

4.1. Droplet relaxation at static equilibrium

When an immiscible cubic 'droplet' fluid is immersed in fluid domain (in the absence of gravity), surface tension will force the formation of the spherical equilibrium shape. The force balance between surface tension and capillary pressure should converge to an exact solution of zero velocity field. The corresponding pressure field should jump from a constant value p_0 outside the droplet to a value $p_0 + 2\sigma/R$ inside the droplet. Modelling the relaxation process of an oil droplet (D_0= 30 μm) in water at static equilibrium serves as an initial demonstration case for testing the suggested methodology, at a mesh resolution of (60x60x60). The fluid properties of the background phase (water) density ρ_1 is 998 kg/m^3 , and the viscosity ν_1 is 1.004e-6 m^2/s, while the droplet phase (oil) densityρ_2 is 806.6 kg/m^3, and the viscosity ν_2 is 2.1e -6 m^2/s, and surface tension of 0.02 kg/s^2.These values result to ($\Delta Pc = \frac{2\sigma}{R} = 2666Pa$). The calculation set up includes a single cubic fluid element patched centrally to the computational domain and it is allowed to relax to a static spherical shape as shown in Fig. 2. It has been shown in the literature [52] that under these conditions and depending on the accuracy of the interface tracking/capturing scheme, non-physical vortex-like velocities may develop in the vicinity of the interface and can result in its destabilization. Tables 1 and 2 demonstrate the different controlling parameters that have been tested. The main testing parameters shown

in the table are: (i) the flux filtering percentage U_f as presented in Eq. 29, (ii) the number of smoothing loops n as presented in Eq. 20, (iii) the sharpening coefficient C_{sh} as presented in Eq. 24 and finally (iv) the compression coefficient $C_{compr.}$ as presented in Eq. 12. Each series of test cases is designed to examine the effect of the mentioned models on parasitic currents and pressure jump calculation accuracy. Cases (S) examine the effect of smoothing loops number in the absence of interface sharpening and filtering. Cases (A) are designed to study the effect of error filtering percentage in the absence of smoothing loops and interface sharpening. Cases (B) examine the combined effect of filtering and smoothing in the absence of interface sharpening, while cases (SE) and (SF) are designed to test the combined effect of smoothing and filtering in the presence of interface sharpening and interface compression, respectively. The adaptive compression scheme introduced in the previous section, is not activated in this case in order to investigate the effect of different pre-specified compression levels on the parasitic current development.

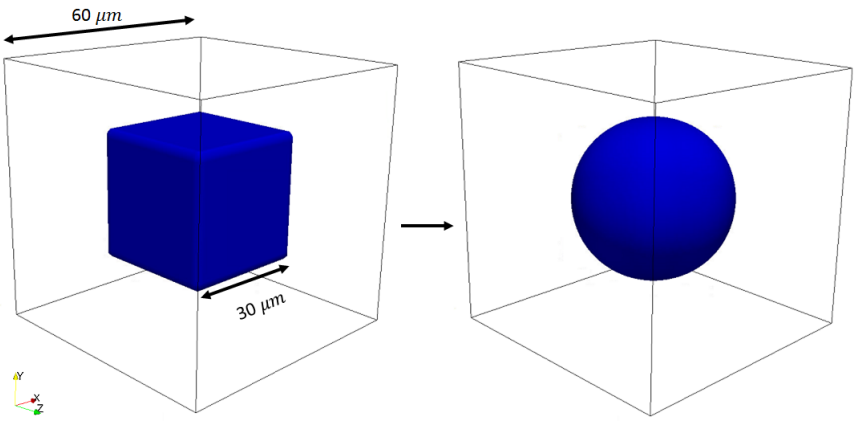


Figure 2: Computational domain for modelling static droplet, (left) initial condition a cube of size D_0 = 30 μm, and (right) static shape of droplet. Mesh size R/δx = 15 at t = 0.0025 s.

	U_f%	n (Eq. 20)		U_f%	n	Filter	U_f%	n (Eq. 20)
Case S1	0	2	Case A1	0.01	0	Case B1	0.05	2
Case S2	0	5	Case A2	0.05	0	Case B2	0.05	5
Case S3	0	10	Case A3	0.1	0	Case B3	0.05	10
Case S4	0	20	Case A4	0.2	0	Case B4	0.05	20

Table 1: Case set-up testing the influence of smoothing and capillary filtering values (U_f% and n) without the effect of sharpening or compression coefficients (C_{sh} and C_{comp} are set to zero)

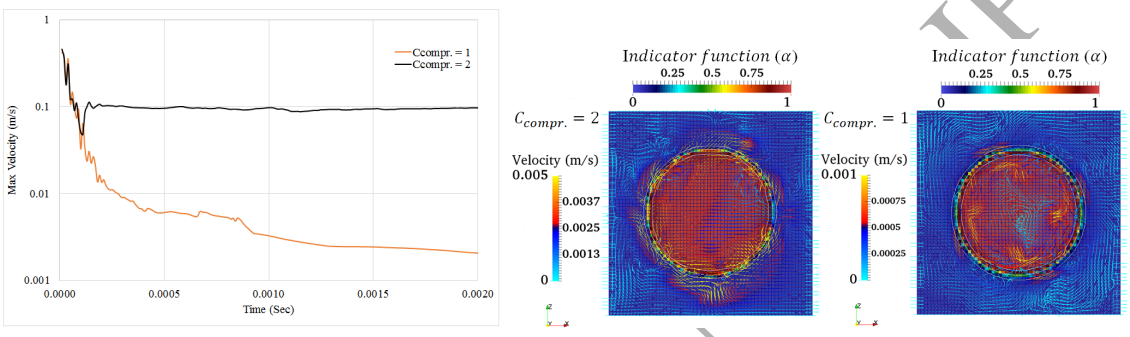
The maximum velocity magnitude in the computational domain is presented as a function of various numerical parameters. If inertial and viscous terms balance in the momentum equation then parasitic veloc-

	$U_f\%$	n (Eq. 20)	C_{sh} (Eq. 24)	C_{comp}
Case SE1	0.05	10	0.1	0
Case SE2	0.05	10	0.5	0
Case SE3	0.05	5	0.1	0
Case SE4	0.05	5	0.5	0
Case SF1	0.05	10	0.5	0.5
Case SF2	0.05	10	0.5	1
Case SF3	0.05	10	0.5	2
Case SF4	0.05	10	0.5	3

Table 2: Case set-up testing the influence of smoothing and capillary filtering values ($U_f\%$ and n) including the effect of sharpening or compression coefficients

ities should be zero. However, the CSF technique introduces an unbalance by replacing the surface force by a volume force which acts over the small region surrounding the continuous phase interface. The surface force suggested by Brackbill et al. [8] includes a density correction as $1/(We\frac{\rho}{\langle\rho\rangle}\kappa n)$ for modelling systems where the phases have unequal density, where ρ is the local density and $\langle\rho\rangle$ is the average non-dimensional density of the two phases. Including these two variables does not affect the total magnitude of force applied, but weights the force more towards regions of higher density. This tends to produce more uniform fluid accelerations across the width of the interface region. Such a force is irrotational and so it can be represented as the gradient of a scalar field. Referring to the momentum equation 2 the surface tension force has to be precisely balanced by the pressure gradient term, with all velocity dependent terms, and thus velocities, being zero. The commonly used VoF numerical implementation of this system differs from this ideal implementation of α, which when discretised represents the volume fraction integrated over the dimensions of a computational mesh cell and varies by a small amount in the radial direction. This results in n-(the normal to the interface) not being precisely directed in the radial direction, κ value varying slightly and the complete interface volume force having a rotational component. The rotational component of the surface tension force cannot be balanced by the irrotational pressure gradient term. So it must be balanced instead by one or more of the three other velocity dependent terms. As these velocity terms (inertial transient, inertial advection and viscous) all require non-zero velocities if they themselves are to be non-zero, spurious currents develop. Looking into the parasitic velocity magnitude for the standard (interFoam) solver during the relaxation period (Fig. 3a), parasitic velocities are high and depend on the compression level. As the value of $C_{compr.}$ increases, the maximum velocity also increases. This might appear to be counter intuitive

since increased compression should result in sharper interfaces, nevertheless, in this work the smooth α field is only used for accurate curvature calculation, but for the rest of the equations the sharpened field had been used curvature κ and the normal vectors. However the sharper the interface the more numerical challenging becomes the calculation of derivatives. Fig. 3a indicates this paradox while Figure 3b presents a graphical explanation. It can be seen that as $C_{compr.}$ increases then vortex like structures develop randomly around the interface that prevent the droplet from relaxing to equilibrium.



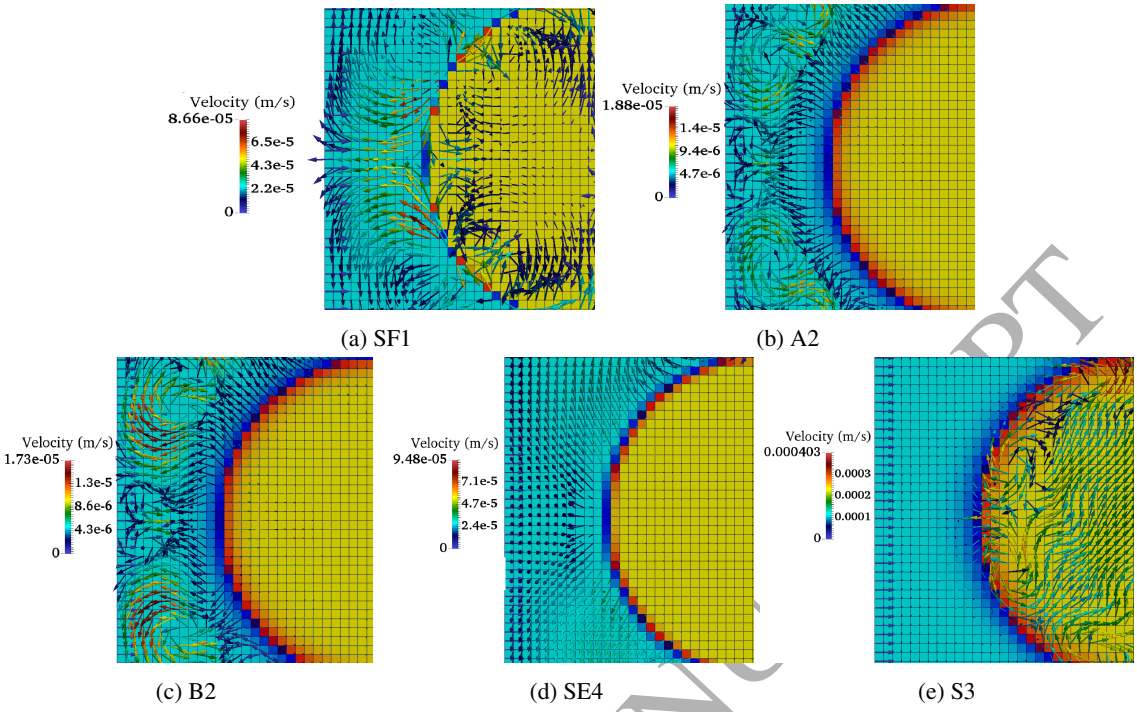
(a) (b)

Figure 3: (a)Evolution of maximum velocity during droplet relaxation using the standard (interFoam) solver with two different interface compression ($C_{compr.}$).(b) values Snapshot of the interface shape after the relaxation of the oil droplet using the standard (interFoam). Velocity vectors near to the interface for different interface compression values are presented.

Testing the smoothing effect presented in Eqs. (19, 20 and 21) using the modified solver by varying the number of smoothing loops (n) as shown of Table (1) is also performed in the presented sub-section. The mentioned set-up in cases S1,S2,S3,S4 is used to investigate the effect of smoothing loops on the parasitic currents, isolated from the other examined controlling parameters. It is evident from Fig. 4e that by increasing the number of smoothing loops, the magnitude of the parasitic currents decreases. However, it should be pointed out that this reduction of parasitic currents, comes at the cost of a corresponding increase in the interface region thickness. Increasing the smoothing loops to 20, the interface thickness increases almost 4 times (6 cells) and parasitic currents tend to develop again and increase by time at a certain point after the relaxation of the droplet. The effect of varying the coefficient U_f for filtering the capillary forces parallel to the interface (see Eq. 30) is revealed from cases A1 to A4 of Table 1; a decrease of the parasitic currents due to the wrong flux filtering near to the interface can be noticed. In the absence of smoothing loops and just changing the filter value U_f, a significant decrease of the parasitic currents is observed as shown in Fig. 4b. Moreover, an optimum decrease in parasitic currents using a value of

$U_f = 0.05$ is observed (Table 1). The decrease of parasitic currents magnitude in this case is a combination of the interface treatment of Eq. 19 and the flux filtering without any smoothing loops being performed. Looking at Fig. 4b one can observe the asymmetric distribution of the velocity vector field with almost zero velocity inside the droplet. By examining the isolated filtering coefficient U_f and smoothing loops n, the suggested framework has been noticed to reduce the spurious velocities, by almost four orders of magnitude, over a relatively long period. Cases B1 to B3 of Table 1 reveal the effect of combining both techniques (smoothing and flux filtering) for damping the parasitic currents; one of the parameters has kept constant - in this case, U_f. Comparing cases (B2) presented in Figures 4c with the previously presented cases S and A, a major improvement in velocity reduction can be seen. In Fig 5 (B) a reduction of almost four orders of magnitude, when compared with the standard solver, has been achieved. By examining the deviation from the theoretical results compared to the standard interFoam using filtering and smoothing models as shown in Table 3, the suggested models reduce the maximum velocity field as seen in cases (S2 and A1), then it start to increase, due to the excessive interface smoothing or the un-balanced capillary forces. Selecting the best smoothing and the filtering coefficient combination ($5 < n < 10$ and $U_f = 0.05$), the effect of the sharpening model Eq. 24 is now examined. In Table 2 cases (SE1 to SE4), the C_{sh} has been varied. Looking at Fig. 4a, a great reduction in the interface thickness can be seen reaching almost one grid cell. By combining the effect of sharpening, filtering and smoothing techniques, the same order of magnitude for parasitic currents with a significant decrease in interface thickness has been achieved. It has also been found that in SF1 case specifically, a very good balance in the velocity vector field with zero velocity inside the droplet (Fig. 5) has been achieved.

As mentioned before, the literature review has revealed the negative effect of increasing the value of compression coefficient, since as the value of $C_{compr.}$ increases the magnitude of parasitic currents also increases. Using the same droplet test case, the effect of increasing the $C_{compr.}$ value on the parasitic current is demonstrated, but this time after applying the smoothing and flux filter models. It should be noted, the aforementioned adaptive compression model is not tested in this case yet, as it will be tested in the next section. In Table 2 cases (SF1 to SF4), the cases using the best combination of the previously mentioned smoothing and filter values coefficient are used with different compression values. The overall maximum velocity values are higher compared to those archived using no compression; nevertheless, these are still



(a) SF1 (b) A2

(c) B2 (d) SE4 (e) S3

Figure 4: Effect of varying model coefficients described in table 1 and 2 on parasitic currents, all figures are showing velocity vector field at t =0.0024 sec. Figures are coloured with indicator function α_{Sharp} as yellow showing oil phase inside the droplet and bright blue showing water outside the droplet

lower than those achieved using the standard solver. A swirling behaviour around the external diagonal direction of the droplet had been noticed as shown in Fig. 4b and 4c. The observed small swirling velocity confirms that the unbalanced surface tension force may increase parasitic currents at one specific location due to this swirling behaviour around the droplet interface. At the same time the effects of the smoothing and the filtering can have positive effect on decaying these swirling velocities.

The behaviour of the droplet when different parameters are considered is important in assessing the impact that the parasitic currents have on the results. Similar simulations but with varying domain sizes (not included in this study) showed that when the parasitic currents were inertia-driven at the deformation phase they spread further across the computational domain. Depending on the nature of the simulation being considered, this may mean that inertia-driven parasitic currents have a greater impact on the results. Quantifying this effect would be difficult, as any integral measure of the parasitic currents – such as the total kinetic energy within the domain for example – would be dependent on additional geometrical factors, such as the domain size and interfacial area. While the form of the velocity field is changing with time

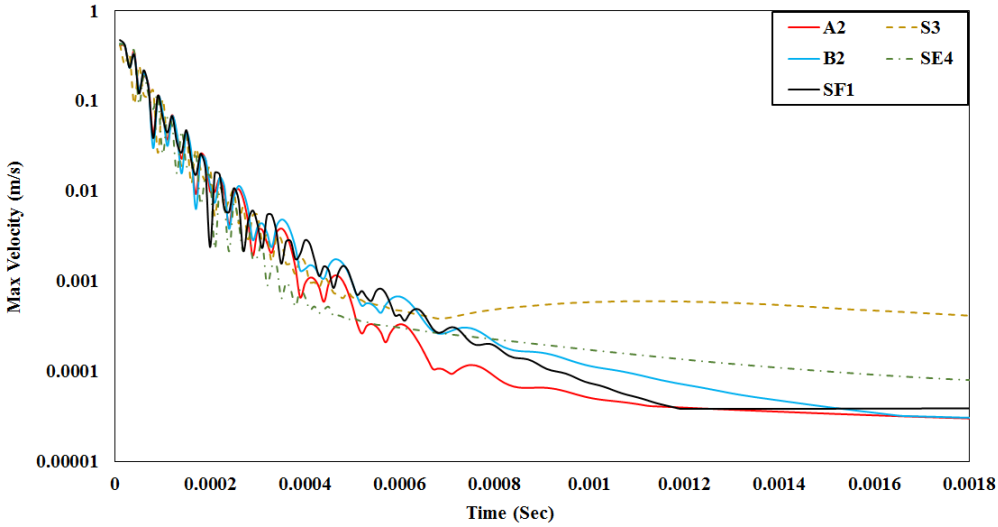


Figure 5: Effect of varying models coefficients presented in table 1 and 2 on maximum parasitic currents over period of time

one can conclude that the parasitic currents are dominated by inertia. The assessment of the effect of different parameters on the maximum velocity can also be presented in the percentage of divergence from the standard solver results as illustrated by Eq. 31;

$$E_{parasitic} = \frac{min(U)}{min(U)_{C_\alpha=2}} \tag{31}$$

where $E_{parasitic}$ represents the error calculated by the $min(U)$ to be the minimum velocity in the domain achieved using modified solver and $min(U)_{C_\alpha=2}$ to be minimum velocity using standard solver at $C_{compr.} = 2$ during the droplet relaxation over a long time interval. Table 3 shows that the magnitude of parasitic currents decreases to minimal in case (B2) where compression and sharpening are null; one can also achieve the same level of reduction in parasitic currents after applying sharpening, as in case (SE3) and with only a slight further increase by adding compression as in case (SF1). Table 3 shows numerically predicted pressure difference between the relaxed spherical droplet and the ambient liquid along the droplet diameter axis for each of the 20 simulated cases, in comparison with the theoretical value predicted from the Laplace equation [see [53] for more details]. The results are presented in terms of the errors in predicted capillary pressure, $Error_{P_c}$, defined as follows:

$$Error_{p_c} = \frac{p_c - (p_c)_{theoretical}}{(p_c)_{theoretical}} / \left(\frac{P - P_{theoretical}}{P_{theoretical}}\right)_{interFoam_{c_{alpha=2}}} \tag{32}$$

where p_c is the calculated capillary pressure using the developed solver, and the P is the calculated pressure using the standard interFoam with compression value of two. The $Error_{p_c}$ presents the deviation of the calculated capillary pressure using the developed solver and the standard solver with respect to the theoretical capillary pressure. Equation 32 shows the reduction in error between the developed solver and the standard solver using compression ($C_{compr.} = 2$). In all the presented cases, reduction in predicting the capillary pressure by 40% can be seen.

$Smooth$	S1	S2	S3	S4
$Error_{p_c}\%$	41.43	40.57	39.64	33.38
$E_{parasitic}$	0.0051	0.0053	0.0080	0.0112
$Filter$	$A1$	$A2$	$A3$	$A4$
$Error_{p_c}\%$	45.55	45.51	45.51	45.63
$E_{parasitic}$	0.0031	0.0006	0.0011	0.0014
$Filter$	$B1$	$B2$	$B3$	$B4$
$Error_{p_c}\%$	44.36	43.39	42.20	40.91
$E_{parasitic}$	0.0005	0.0006	0.0013	0.0032
$Sharp$	$SE1$	$SE2$	$SE3$	$SE4$
$Error_{p_c}\%$	43.04	45.14	43.97	46.11
$E_{parasitic}$	0.0008	0.0024	0.0007	0.0015
$Sharp$	$SF1$	$SF2$	$SF3$	$SF4$
$Error_{p_c}\%$	49.79	50.20	50.12	49.95
$E_{parasitic}$	0.0008	0.0045	0.0057	0.0067

Table 3: Reduction in predicted capillary pressure and parasitic currents compared to the standard interFoam

4.2. Interacting Parasitic Currents of two relaxing droplets

In this section the effect of parasitic current interaction for the case of two stagnant droplets that undergo the same relaxation process is discussed. The same droplet properties as in the previous test case have been used (see Section 4.1). When two droplets are found in the same domain in close proximity, the parasitic currents may interact resulting in artificial movement of the droplets and eventually merging. Figure 7 shows the velocity magnitude on the droplet represented by the 0.5 liquid volume fraction iso-surface. The same set of parameters are utilised as in (A2, B2, SE3 and SF1) cases mentioned in Tables 1 and 2. One can notice in Fig. 7a to Fig. 7c that the two droplets have merged to one big droplet located at the centre of

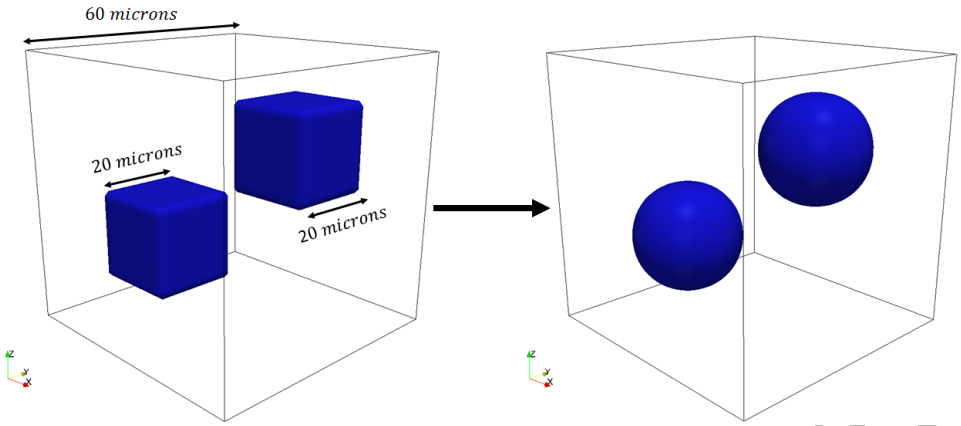


Figure 6: Computational domain showing two static droplets , (left) initial condition a cube of size $D_0 = 20\ \mu$m each, and (right) static shape of droplet as two boxes.

the computational domain. In contrast Fig. 7d shows that the two droplets remain in their initial position as they should. This can be considered as a demonstration that optimising compression for one case does not necessarily mean that can offer optimum results for other similar cases and the solver should automatically adapt the needed compression. Hence, in the next sections that consider cases with higher deformation of the interface we are going to introduce the adaptive solver.

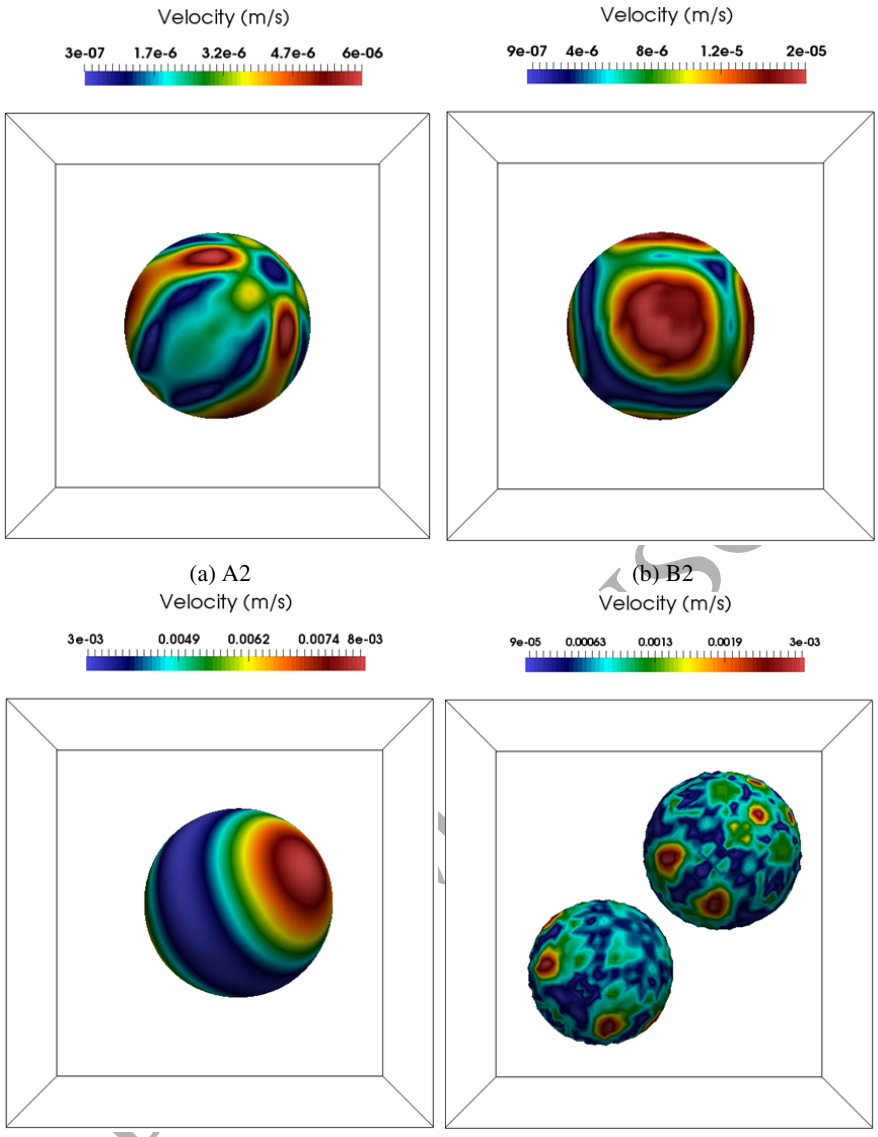
4.3. Notched disc in rotating flow

In addition to the static droplet test cases, the rotation test of the slotted disk, which is known as the Zalesak problem [51] has been tested. The Zalesaks circle disk is initially slotted at the centre (0.5,0,0.75) of a 2D unit square domain. The disk is subjected to a rotational movement under the influence of a rotational field that is defined by the following equations:

$$u(x) = -2\pi(x - x_0) \tag{33}$$

$$w(z) = 2\pi(z - z_0) \tag{34}$$

where u(x), w(z) are the imposed velocity components. By applying this velocity, one complete rotation of the disk is completed within t = 1sec. For all simulations performed for this test case, a fixed time-step has been used, keeping the Courant number equal to 0.5. The initial disk configuration used for the simulation



(a) A2 (b) B2

(c) SE3 (d) SF1

Figure 7: Effect of combined flux filtering and smoothing in the presence of sharpening model on the interaction of parasitic velocity field. All figures are showing the velocity field at t =0.0024 sec on the indicator function α_{Sharp} iso-contour = 0.5

is presented in Fig. 8. Three different mesh densities were used consisting of 64x64, 200x200 and 400x400 cells, respectively.

Figures 9 and 10 show the comparison between the standard solver using different compression ($C_{compr.}$) values and the developed adaptive solver using different sharpening (C_{sh}) values. In each plot, the exact initial and final interface shape is presented. In all the figures, the iso-contours values of indicator function alpha α of (0.1, 0.5 and 0.9) after one revolution of the disk are shown. The reason of presenting three

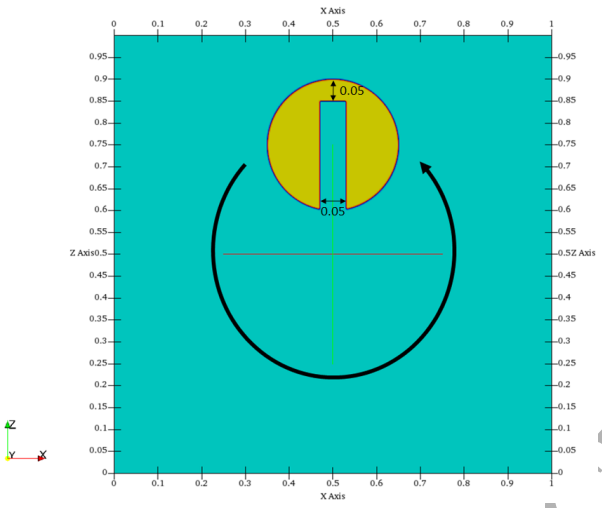


Figure 8: Schematic representation of two dimensional Zalesak's Disk benchmark test case described at [54].

contour lines is to better explore the effect of the adaptive compression model on both the interface diffusion and the overall disk shape. For the coarse mesh (64x64) neither using the standard interFoam with three compression values ($C_{compr.}$ = 0, 1 and 4)), nor the three values for C_{sh}, (C_{sh} = 0.1, 0.5 and 0.9) for the adaptive modified solver, can provide a satisfactory interface representation. One can even notice that due to the large interface deformation and diffusion, the interface iso-contour of $\alpha = 0.9$ at Fig. 9(a) has disappeared for the standard solver. Nevertheless, for the adaptive modified solver cases, the modified solver can keep the main geometrical features as seen in Figs. 10(a,d,g). By using high compression as in Fig. 9(g) , one can notice a reduction in the interface thickness, although a rather high deformation and corrugated shape of the final disk shape has been noticed. Comparing Fig. 9(g) to Fig. 10(g) one can notice the effectiveness of the adaptive model that preserves the geometrical outline of the disk while the sharpening model decreases the interface thickness. Moving to a finer mesh (200x200), high interface diffusion using the standard interFoam with no compression ($C_compr.$ = 0) Fig. 9(b) has been noticed. The higher grid resolution is not adequate to provide remedies to the previously mentioned deficiencies noticed in the coarser mesh using interFoam. The highly diffusive interface using the standard interFoam also did not maintain the 0.9 iso-contour making two oval shapes at the sides. For higher compression values Fig. 9(e,h) although the disk shape is preserved by the standard solver, the interface is significantly deformed near the outer disk boundary. Use of the adaptive solver Fig. 10(b,e,h) shows better consistency for the shape regardless of the imposed sharpening level. Moreover, the adaptive compression eliminates any irregular

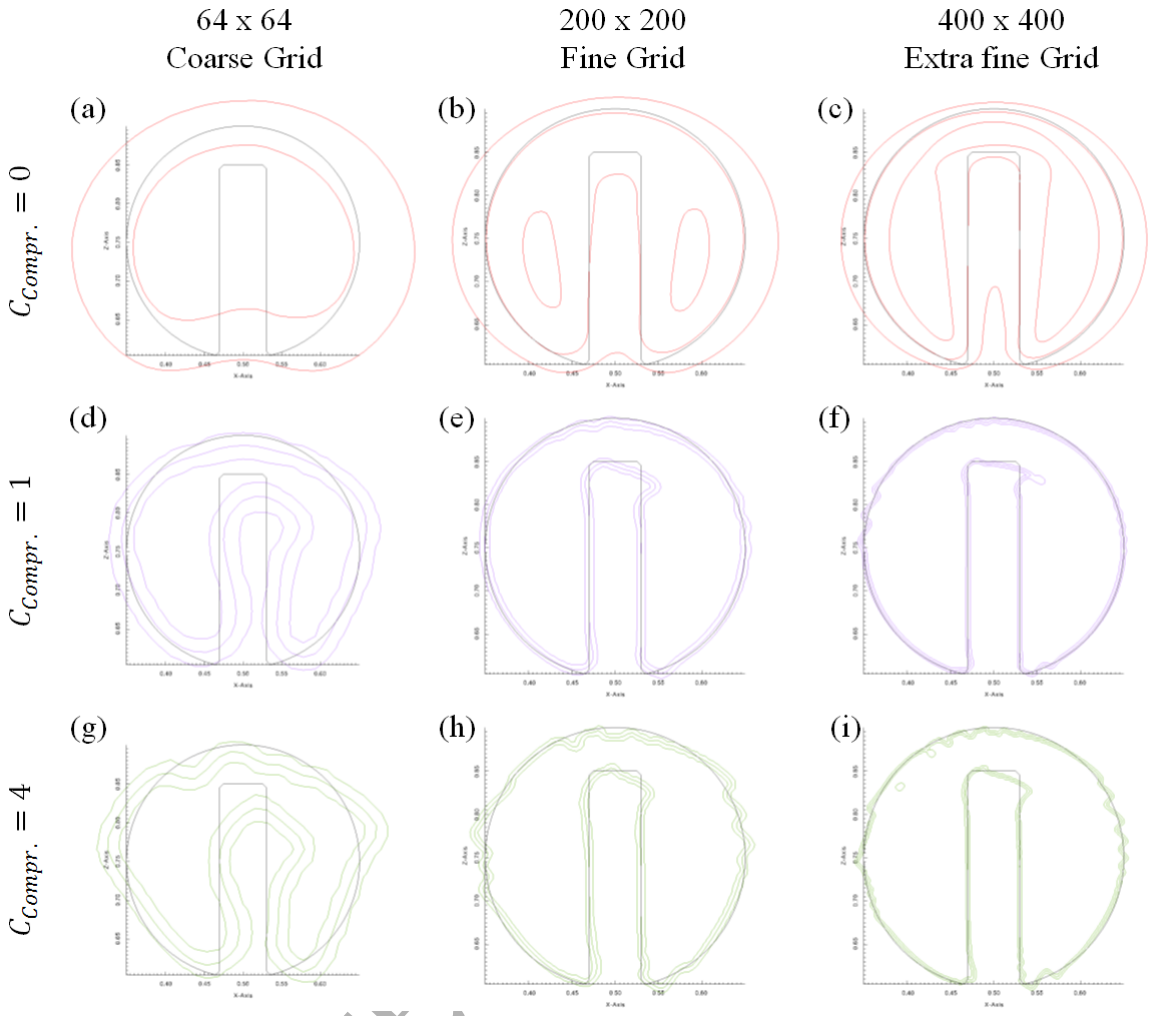


Figure 9: Zalesak disk after one revolution. Iso-contours for indicator function alpha (α = 0.1, 0.5 and 0.9) are plotted for the standard interFoam using different compression values, together with the reference shape.

shapes compared to the standers solver. Figure 10(h) especially shows an excellent agreement with the original circular shape layout. This test case also demonstrates the role of the sharpening value C_{sh} which can help in controlling the interface diffusion depending on the case under consideration. To examine our adaptive solver mesh dependency, the mesh has been doubled to 400x400. Even for this fine grid resolution case the standard solver gives inaccurate disk shape regardless of the compression value used, as none of them is adequate to balance the interface shape. A zero compression value using the standard interFoam preserves the characteristic shape for the first time (see Fig. 9(c), compared to Fig. 9(a,b)). For the higher compression values as in Fig. 9(f,i), high corrugated regions at the interface have been observed. Using the

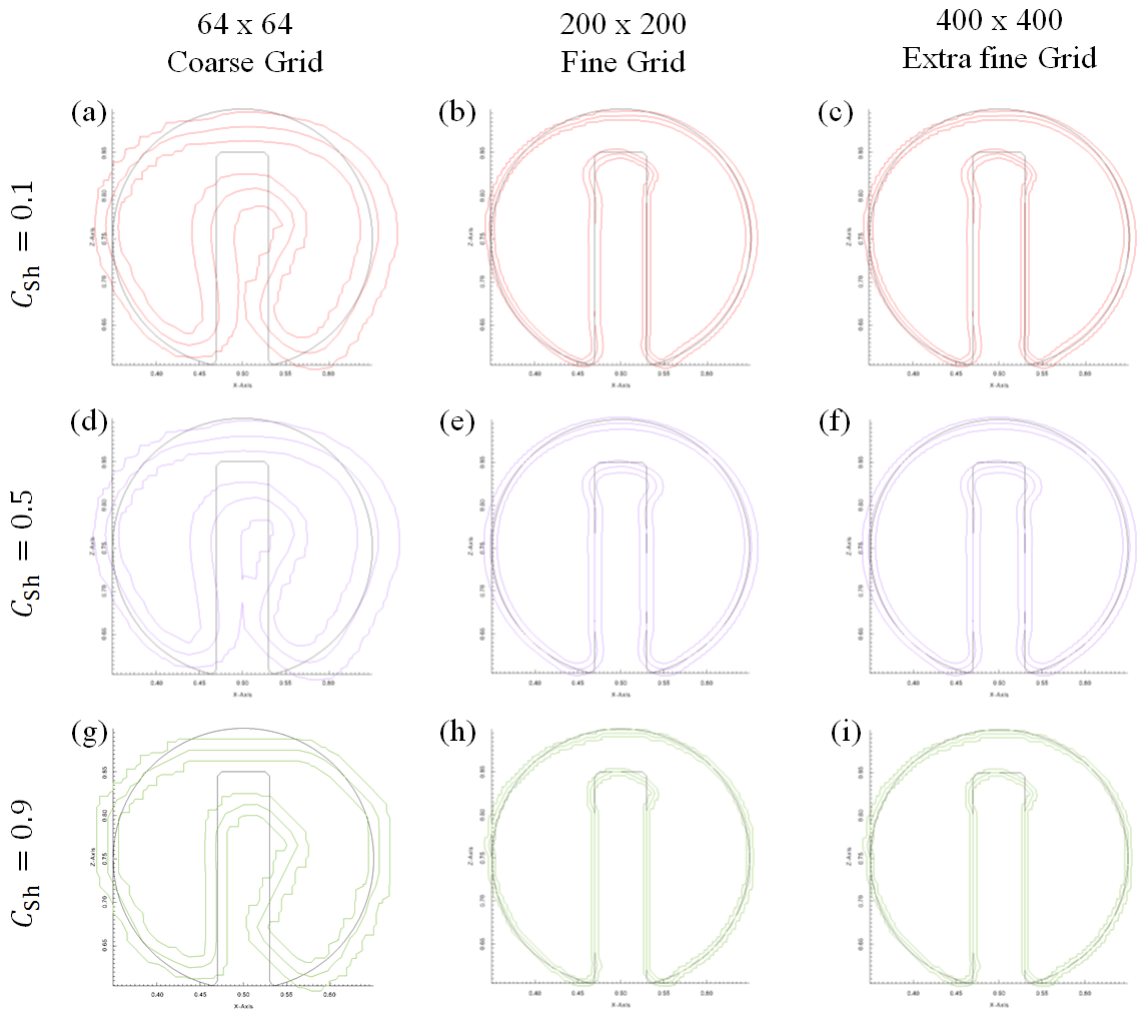
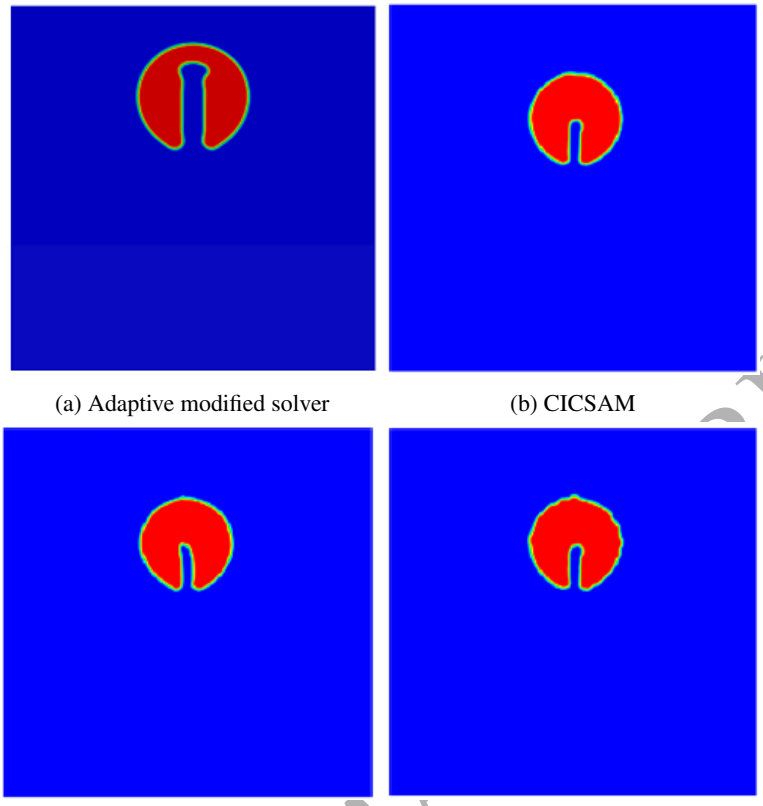


Figure 10: Zalesak disk after one revolution. Iso-contours of indicator function alpha sharp (α_{Sh} = 0.1, 0.5 and 0.9) are plotted for the adaptive modified solver using different sharpening coefficients, together with the reference shape.

adaptive modified solver a better disk shape representation has been obtained, regardless of the sharpening coefficient value C_{sh} (see Fig. 10(c,f,i)). Moreover, by using the three different sharpening coefficients C_{sh} a thickness of approximately 1-2 cells has been preserved. Also a minimum difference between the fine and the extra fine grid in terms of interface thickness has been observed, and sharpening algorithm shows the perfect fit to the internal notch. These observations indicate that adaptive compression is less sensitive to tuning parameters such as the sharpening (see Eq. 24), which is not effective for coarse grid resolution.

For completeness, results included in [20] are also shown. In [20] various commonly used interface capturing methods have been presented for the same test case; these include the standard compression



(a) Adaptive modified solver (b) CICSAM

(c) PLIC (d) FCT

Figure 11: Comparison between the used framework and available method reviewed by Gopala and van Wachem [20]. (a) is showing modified solver with adaptive compressive scheme, (b) is showing the compressive interface capturing scheme for arbitrary meshes (CICSAM), (c) is showing piecewise linear interface construction (PLIC) and (d) is showing flux-corrected transport FCT. All presented in mesh a domain of 200 by 200

scheme used by OpenFOAM, the compressive interface capturing scheme for arbitrary meshes (CICSAM) employed by FLUENT commercial code, the piecewise linear interface construction (PLIC) and the flux-corrected transport (FCT)). In this test cases, the notched disk was a bit different than what is presented in the standard Zalesak [51] test case, yet it has the same overall characteristics. Looking at this comparison, one can relate and compare the overall behaviour for the different solvers as seen in Fig. 11. Nevertheless, one can spot out the difference in geometrical layout between our test case and the test cases presented in [19]; the mesh was kept the same as in [20] (200x200). By comparing the results from the developed solver to those reported in [20], it can be concluded that a good solution has been achieved.

4.4. Circle in a vortex field

In this section, the solver performance is tested in a vortex flow as presented by Rider and Kothe [42] and Roenby et al. [34]. The aim of this benchmark test is to verify the ability of the model to deal with severe interface stretching. The test case includes an initially static circular fluid disk with radius of R = 0.15 mm centred at (0.5,0,0.75) in a unit square domain. The disk is subjected to a vortex as shown in Fig. 12. The axis of rotation is located in the centre of the field, and can be described by the following stream function;

$$u(x, z, t) = \cos((2\pi t)/T)(-\sin^2(\pi x)\sin(2\pi z), \sin(2\pi x)\sin^2(\pi z)) \tag{35}$$

where u is the field rotational velocity and T is the period of the flow during rotation. Due to the flow direction, the disc is stressed into a long thread until time $t = 4s$ forming a spiral shape. The interface thickness of the deformed disk shape, as well as the numerical diffusion of values located at the tail of the fluid body during its spiral motion are of interest. The results presented in Fig. 13 and 14 are for three different grid sizes using the standard (interFoam) and the newly developed adaptive modified solver. On each figure, the final interface shape is shown with three iso-contours values for the indicator function (α) of (0.1, 0.5 and 0.9) after one revolution of the disk (t= 4 s).

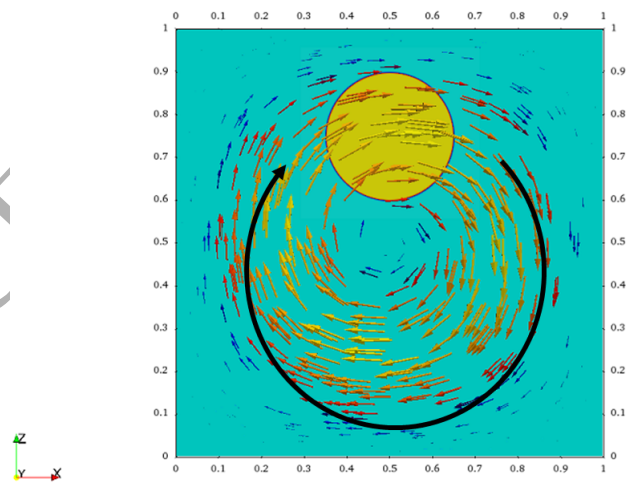


Figure 12: Schematic representation the initial configuration of the shearing flow test with the value of the color function is one inside the circle and zero outside

The standard solver failed to capture the full spiral shape after the disk rotation using the coarse mesh

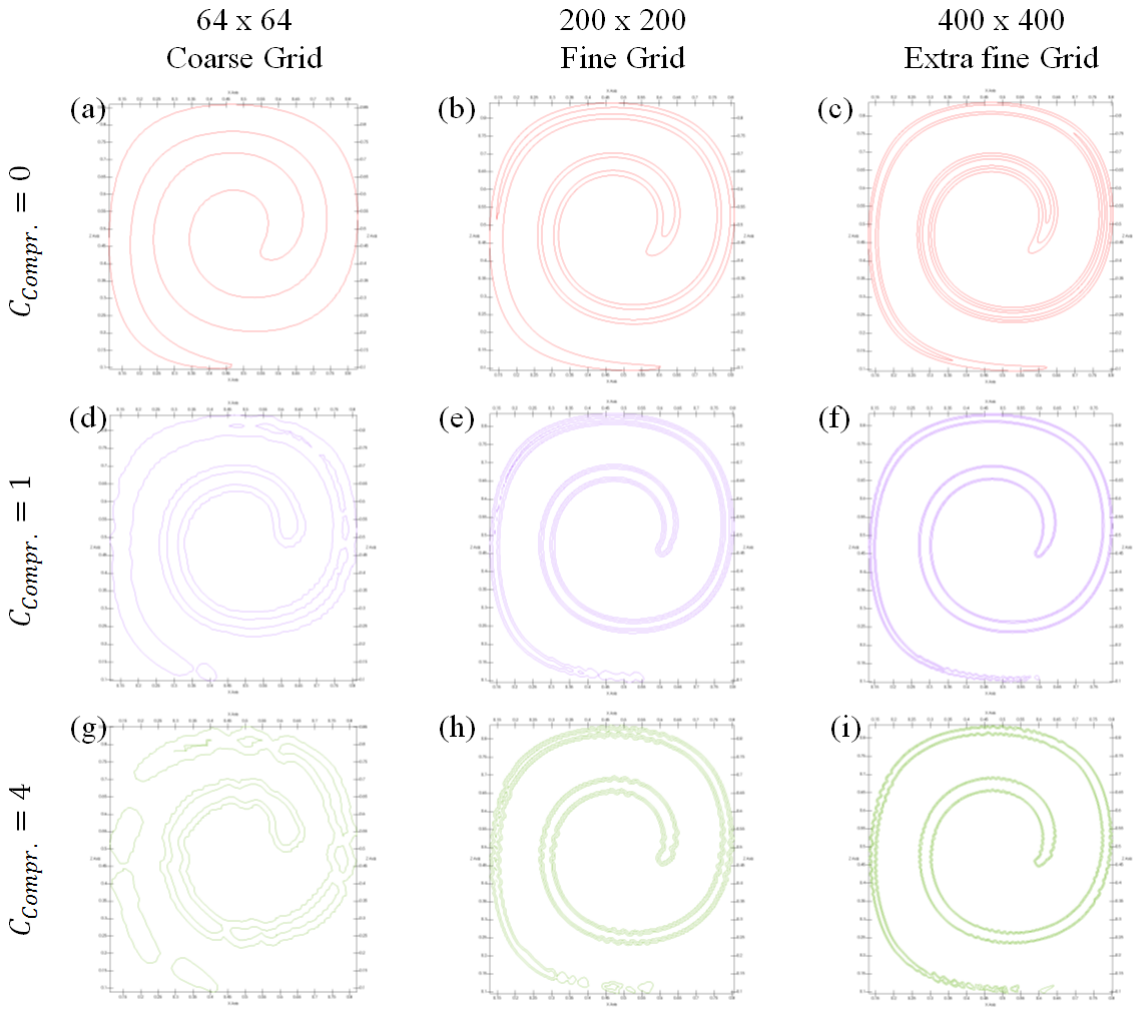


Figure 13: Circle in a vortex field after one revolution. Iso-contours for indicator function alpha (α = 0.1, 0.5 and 0.9) is plotted for the standard interFoam using different compression values, together with the reference shape.

(see Fig. 13(a,d,g)). Due to the very high diffusion and the absence of compression, iso-contours of 0.1 and 0.5 volume fraction have disappeared from the computational domain (see Fig. 13 (a)). Using the adaptive modified solver the results are problematic as well especially for the tail as presented in Fig. 14(a,d,g). By using high sharpening value Fig. 13 (d,g) at low grid resolution to counter balance the numerical diffusion, tail snap-off at the spiral formation has been observed. Fragmentation or tail snapping off is evident in all figures.

Moving to a finer grid (200x200) the behaviour of the two solvers becomes similar although some differences can be noticed. The standard solver with no compression Fig. 13(b) suffers from high diffusion as seen

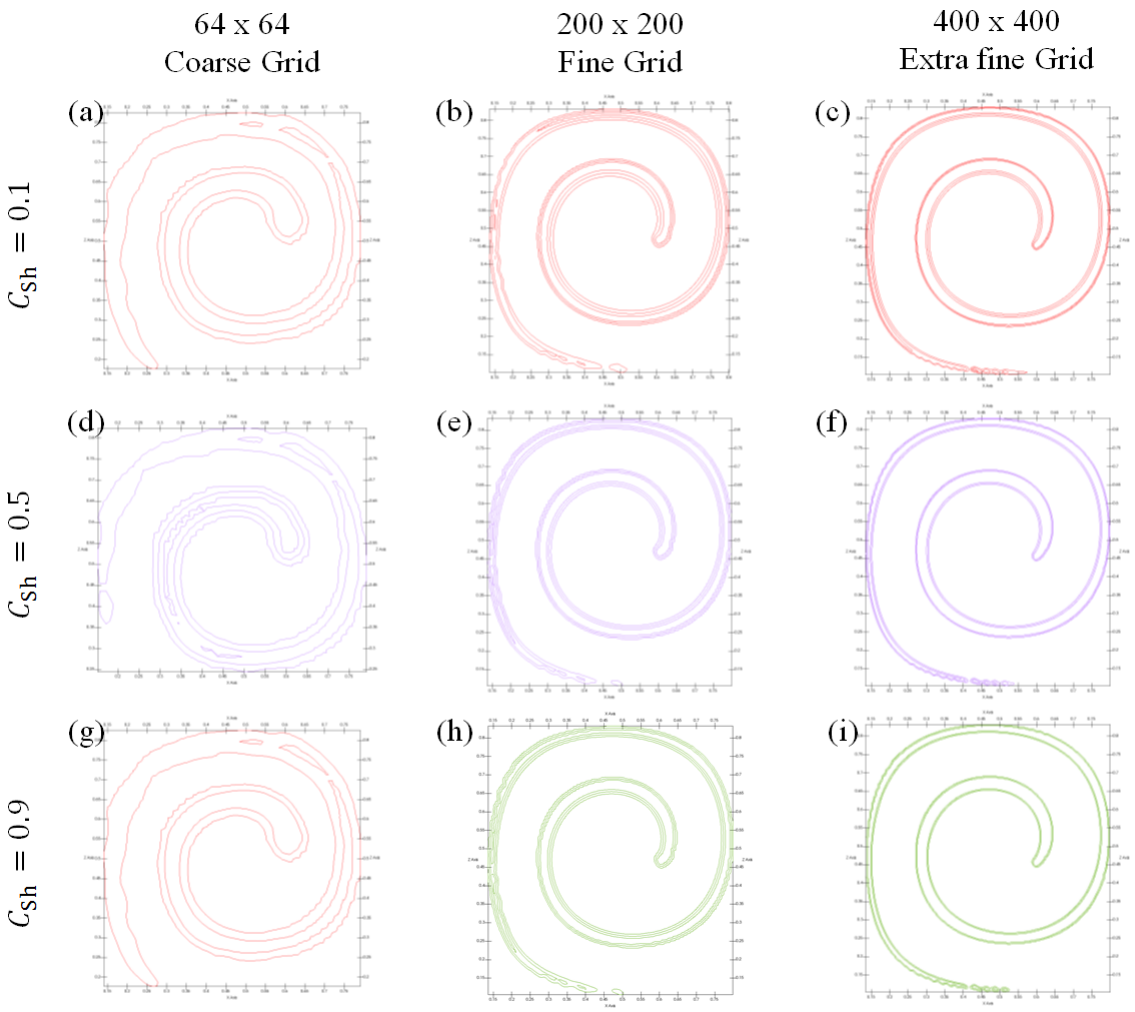


Figure 14: Circle in a vortex field after one revolution. Iso-contours of indicator function alpha sharp (α_{Sh} = 0.1, 0.5 and 0.9) is plotted for the adaptive modified solver using different sharpening coefficients, together with the reference shape.

in the previous test cases where the (0.1) iso-contour disappears. As the compression value increases (see Fig. 13(e,h)) the standard solver shows early fragmentation at the tail or non-smooth interface. In contrast, the adaptive solver agrees with the expected spiral shape using different sharpening coefficients. Nevertheless, with low sharpening value as shown in Fig. 14(b) early fragmentation with the 0.1 iso-contours lines loss has been observed. Increasing α_{Sh} to values greater than 0.5 (see Fig. 14(e,h)) provides an accurate spiral shape with minimum phase snapping at the tail. Good agreement using adaptive compression has been achieved in balancing the swirling tails compared to the wiggly interface appeared using the standard solver. One can notice that the smallest fragmentation at the spiral tail seems to be unavoidable by using any

applied sharpening algorithm, as also discussed by Sato and Ničeno [55] and Malgarinos et al. [26], especially at regions where the liquid body becomes very thin. Fragmentation happens when the local interface curvature becomes comparable to the cell size. At this point, the iso-contours are not able to represent the significant interface curvature inside the cell any more. Iso-contours based on volume fraction advection, leads to errors in the estimate of the fragmented droplet motion similar to those reported by Černe et al. [56] and Roenby et al. [34]. As a final sensitivity test the grid size has been doubled (400x400), to examine the influence of the mesh size on the adaptive solver. Both solvers perform better with this high resolution grid, yet differences have been noticed as with the previous cases. As seen in Fig. 13(c) the standard (interFoam) using zero compression coefficient gives a better interface representation with less diffusion and stable tail. By introducing compression (see Fig. 13(f,i)) the spiral shape is maintained, although wiggly shapes emerge near the outer interface. Using the adaptive compression no significant change is noticed; by varying the sharpening value (C_{sh}): as seen in Fig. 14(c,f,i), the results do not change. The results indicate that the balance between sharpening and compression is well achieved. Combining the developed solver with fine grid proves the proposed methodology independent of tuning parameters which is a very desirable feature within multiphase flows. Finally, it had been concluded that even by using medium quality mesh (i.e. 200x200), the adaptive solver can provide satisfying results for a wide range of sharpening coefficients.

4.5. Bubble formation at T-junction

The previous benchmark cases tested the suitability of the developed model to a range of idealised conditions. No significant topological changes occur and wettability effect is not present. Thus, further validation against experimental data for the case of formation of bubbles in a T-junction has been performed. This is a test case that involves wetting conditions at the wall as well as complex fluid interface topological changes through the breakup and generation of bubbles. The focus is to test the accuracy of our adaptive model in estimating the correct bubble shape and frequency as presented in the experiment of Arias et al. [57]. Full wetting conditions ($\theta = 0°$) at the main tube are used. Moreover, the contact angle imposed on the injection tube (see Fig. 16) has been taken from the corresponding flow images. A constant contact angle of $\theta = 25°$ for the left wall and $\theta = 45°$ for the right wall has been chosen to match the experiments. The connection between the two channels as well as the flow directions and geometrical representation are shown in Fig. 15.

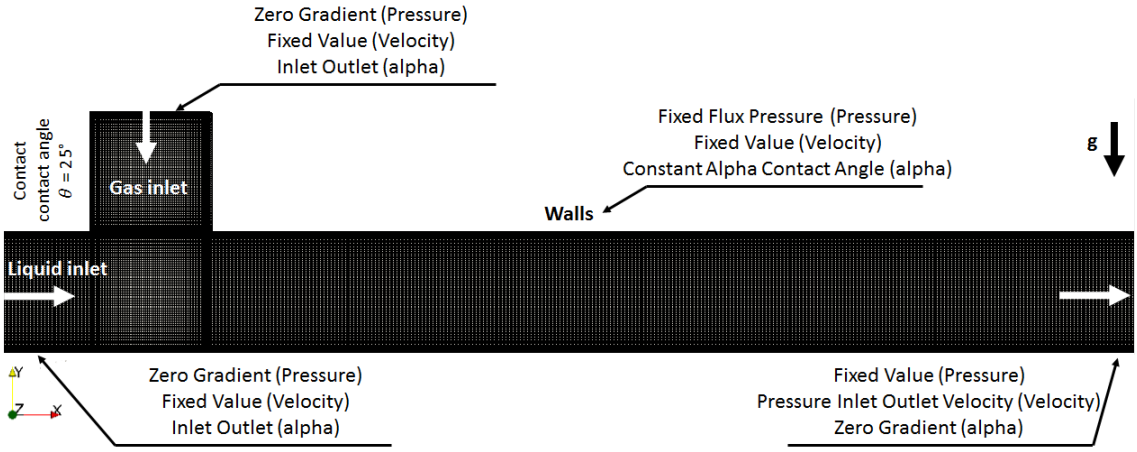


Figure 15: Geometrical model boundaries and overall dimensions

Two different operating conditions, summarised in Table 4, have been selected for presentation. The velocities selected for comparison with our numerical simulations are also shown in table 4. The conditions used are carefully selected to simulate low capillary number and to show two different bubble size formation with fluid properties listed in Table .5.

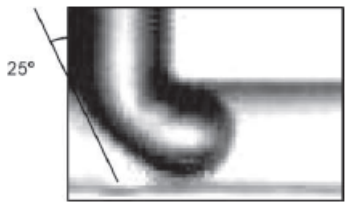


Figure 16: Contact angle at injection tube measured from experimental images

Table 4: Inlet velocities for liquid and gas, dimensionless numbers and regime expected

Case	$U_g(m/s)$	$U_l(m/s)$	$MaxRe$	$MaxWe$	$Exp.Regime$
Case 1	0.242	0.318	32	1.4	$Slug$
Case 2	0.068	0.531	53	3.92	$Bubble$

For this test case the appearance of spurious numerical currents would create instability during the bubble formation process. These currents induce unphysical vortices at the interface, destabilising the simulations and strongly distorting the interface movement. Gravity acceleration constant was 9.8 m/s^2, while the values of maximum Weber number ($\frac{\rho DU^2}{\sigma}$) and the maximum Reynolds number ($\frac{\rho DU}{\mu}$) were the

Table 5: Fluid physical properties

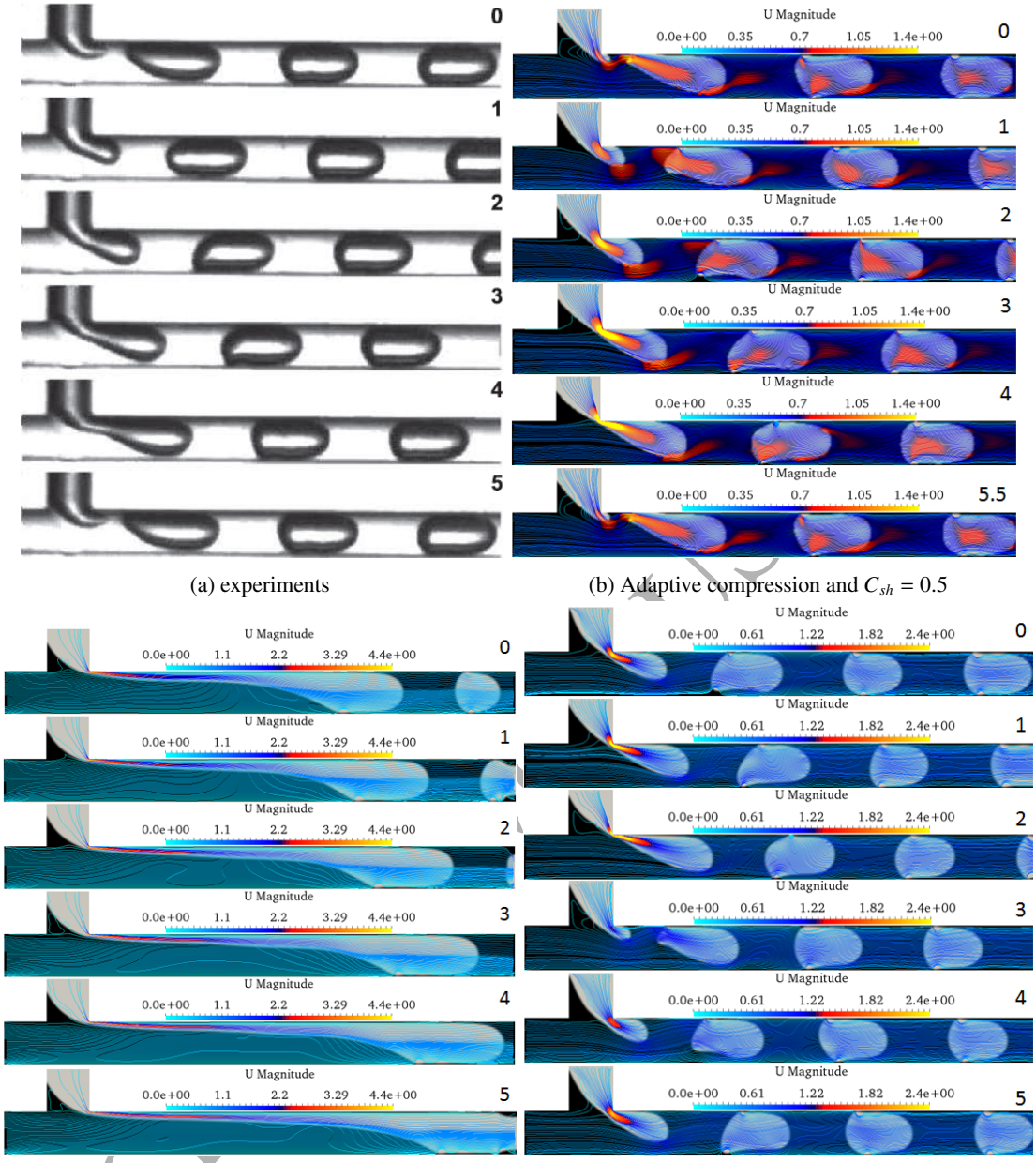
	$\rho(Kg/m^3)$	$\nu(m^2/s)$	$\sigma(N/m)$
Water properties at $25°C$	1000	$1.004x10^{-6}$	0.07
Air properties at $25°C$	1.2	$8.333x10^{-6}$	0.07

same as in the experiments and shown in table 4.

Comparison of the results from the modified solver and the standard solver (interFoam) using different compression values against the experiments are shown in Figs. 17 and 18. Depending on the inlet velocity imposed, one should expect to reproduce different bubbles formation.

Figure 17 presents the first bubble generation sequence as mentioned in case 1 Table 4. Using the standard solver, the slug formation is achieved only when adjusting the compression coefficient to the value of two as seen in Fig. 17d. Even in this case though the detached ligaments of the fluid appear to be more spherical than what the experiments indicate. Using the comparison value of one the standard solver failed to predict the interface snap-off as seen in Fig. 17c. In contrast looking at Fig. 17b it is noticed that the results obtained by the new adaptive model agree very well with the experiments in terms of both slug formation and snap-off time as seen in Fig. 17a. The adaptive framework predicts the interface snap-off correctly and minimises the overall parasitic currents. Moreover, the standard solver shows a considerable increase in parasitic velocity near the interface that may reaches eight times the magnitude of the flow velocity. The new solver achieved low parasitic currents during the snap-off events while maintaining an accurate sharp interface.

Figure 18 presents bubble flow patterns obtained by imposing higher liquid velocity but lower gas velocity as in case 2 Table 4 in comparison to the previous case. Good agreement in terms of shape and patterns between experiments and all numerical simulations can be observed regardless of the solver used. It is worth mentioning though that looking at Figs. 18c, 18d when the standard interFoam solver is used, bubbles are generated at different frequencies based on the compression coefficient value. By comparing the two figures to the experimental Fig. 18a one can also notice that the snap-off time is delayed compared to the experimental results, while in Fig. 18b one can observe that using the developed adaptive solver, the snap-off time and the bubble generation frequency is matching well with the experiences. According to the experimental observations, bubble generation results from the breakup of a gas thread that develops



(a) experiments

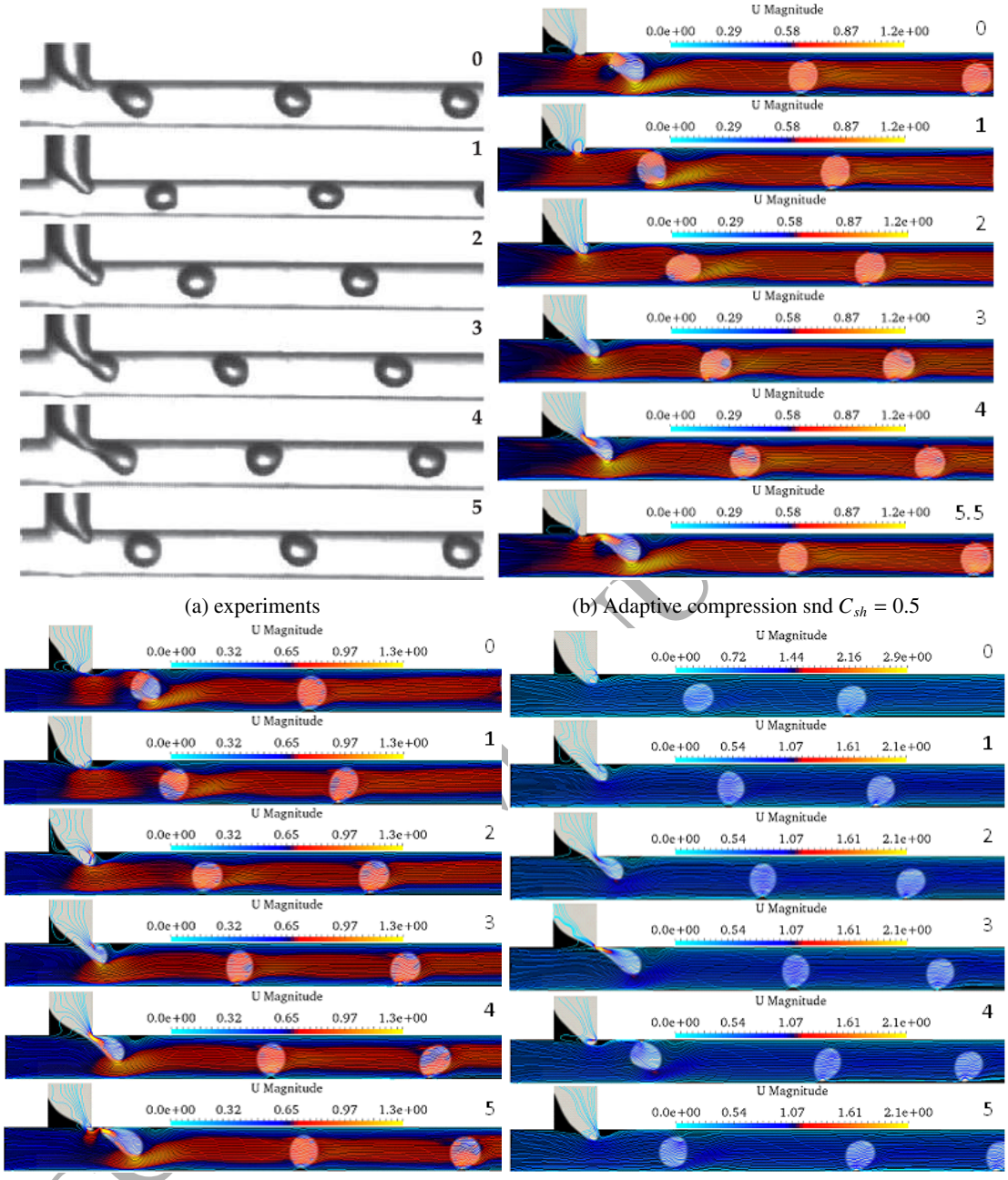
(b) Adaptive compression and $C_{sh} = 0.5$

(c) interFoam $C_{alpha} = 1$ standard solver

(d) interFoam $C_{alpha} = 2$ standard solver

Figure 17: Slug flow. (a) experiments and (b,c,d) numerical simulations. $U_L = 0.318$ m/s and $U_G = 0.242$ m/s. Time (ms) is indicated in the upper right corner. Stream lines are coloured with velocity magnitude in all the figures.

after the T- junction. The explanation for the breakup is supported by the Plateau-Rayleigh instability as discussed by Ménétrier-Deremble and Tabeling [58] or by the effects of the flowing liquid from the tip of the thread to the neck where pinch-off occurs as presented by van Steijn et al. [59]. The surface tension has a stabilising effect and opposes any deformation of the interface tending to create a bubble. The snapping



(a) experiments

(b) Adaptive compression snd $C_{sh} = 0.5$

(c) interFoam $C_{alpha} = 1$

(d) interFoam $C_{alpha} = 2$

Figure 18: Bubble flow, (a) experiments and (b,c,d) numerical simulations. $U_L = 0.531$ m/s and $U_G = 0.068$ m/s. Time (ms) is indicated in the upper right corner. Stream lines are coloured with velocity magnitude in all the figures.

events discussed by the previous literature are in agreement with the simulations presented here,since no unnatural pinch-off has been observed using the modified solver. On the other hand, a long thread of gas generated using (interFoam) is clearly seen in Fig. 17c.

Table 6: Error in Bubble generation frequency

	$Sim.frequency(Hz)$	$Error_f$
Case 1 (Modified solver)	190.47	4.7 %
Case 1 (interFoam C_{alpha} = 1)	210.53	5.2 %
Case 1 (interFoam C_{alpha} = 2)	No Bubble generation	100 %
Case 2 (Modified solver)	200.00	1.9 %
Case 2 (interFoam C_{alpha} = 1)	184.00	9.8 %
Case 2 (interFoam C_{alpha} = 2)	179.21	12.15 %

In the previous section a qualitative comparison has been demonstrated using the standard solver and the developed solver against different variation of the control parameters. The validation has been extended to quantitatively compare the bubble generation frequency with experiments. To ensure regularity in the formation of bubbles, a train of bubbles is generated containing at least four of them. The generation frequency was estimated by measuring the time required to create the bubbles. The first bubble of each train, which was strongly dependent on the initial geometry was not considered. We quantify the accuracy of the bubble generation frequency using the following equation:

$$Error_f = \frac{Sim.freq - Exp.freq}{Exp.freq} \tag{36}$$

where the *Sim. freq* is the time calculated from the simulations in order to generate one bubble and *Exp. freq* is the time needed to produce one bubble in the actual experiment. Table 6 shows the error in the bubble frequency generation compared to the experimental data. For Case (1) although the qualitative results are very close between Fig. 17b and Fig. 17d, one can notice that the developed solver can achieve better accuracy in the in bubble generation frequency. In case (2) the simulation data are qualitatively similar to the experimental results.

5. Conclusions

A multiphase flow solver for interface capturing at low capillary number flows has been developed and evaluated against well established benchmark cases. Wide range of control parameters of the VoF methodology have been tested, aiming to shed light to their effect on physical properties of micro-scale flows as well as how they interlink. Five different test cases, chosen specifically to highlight the strengths and sensitivity of each model are presented; the best results obtained are summarized in Tables (7,8). The

present work was intended to overcome a natural tendency to evaluate numerical methods using only test cases close to the specific application for which they were designed in the first place. In our study a wide range of conditions have been tested, starting from static interfaces (static droplet), and moving to interface smearing (Zalesaks disk, circle in a vortex field) and bubble generation using experimental (T-junction). As it has been demonstrated, although for all the test cases there is a unique optimum set of parameters relevant to sharpening and smoothing part of the method ($U_f\% = 0.05, n = 10, C_{sh} = 0.5$), this is not the case for the $C_{compr.}$ term. The results presented here as well as in previous literature studies, indicate that this term is the most versatile coefficient depending on the physical characteristics of the case under consideration as well as the grid size. With the inclusion of adaptive compression this difficulty is waved and an a-priori selection of a value is not required. Even more importantly, it seems that the adaptive nature of the coefficient that controls the interface thickness counter balances the need for very fine grids. The combination of an adaptive compression VoF algorithm and a smoothing technique for the computation of the surface tension has been shown to give accurate results and satisfactory convergence. Advection tests in which interfaces are transported by an assumed external velocity field have been considered while a quantitative comparison with previous literature has been also made. In addition, bubble formation in a liquid flow was simulated by solving the Navier–Stokes equations coupled to the volume fraction field equation in a T-junction configuration for which experimental data are available. From the advection test cases, where the volume fraction equation is solved, the compression method as implemented in the solver interFoam failed to predict the results qualitatively. In contrast, the results obtained with the adaptive modified solver, adhere closely to literature. The used adaptive compression method proved to be mass conserving. In the future work, the proposed method will be used to model multiphase flow using real porous rocks produced from micro-CT images to characterize the effect of wettability on droplet impacting porous media.

Table 7: Benchmark summary highlighting the best set-up for static droplet test cases, along advantages and disadvantages

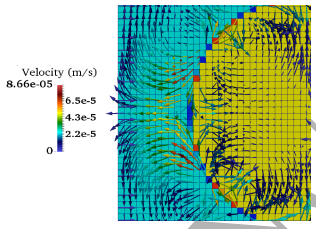
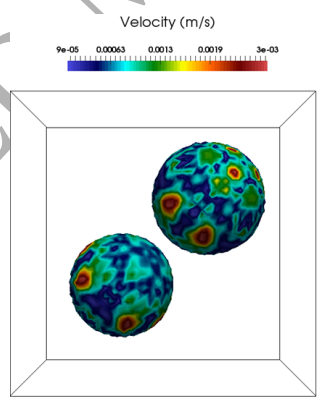
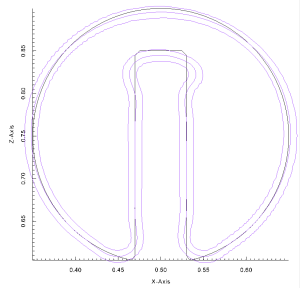
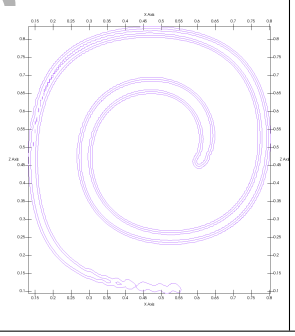
Benchmark	Control parameters	Effective results	Comments
Static Droplet	Case SF1 • $U_f\% = 0.05$ • $n = 10$ • $C_{sh} = 0.5$ • $C_{compr.} = 0.5$		Advantage • Interface presented in one grid cell Disadvantage • Sensitive to compression co-efficient value (C_{comp} tested 0.5,1,2,3) • Adaptive compression not used
Interacting Parasitic Currents of two relaxing droplets	Case SF1 • $U_f\% = 0.05$ • $n = 10$ • $C_{sh} = 0.5$ • $C_{compr.} = 0.5$		Advantage • Interface presented in one grid cell • Droplets do not merge Disadvantage • Sensitive to compression co-efficient value (C_{comp} tested 0.5,1,2,3) • Higher parasitic current than one droplet test • Adaptive compression not used

Table 8: Benchmark summary highlighting the best set-up for a typical advection test cases, along advantages and disadvantages

Benchmark	Control parameters	Effective results	Comments
Zalesaks Disk	Fine Grid (200 x 200) • $U_f\% = 0.05$ • $n = 10$ • $C_{sh} = 0.5$ • $C_{compr.} = Adptive$		Advantage • Not sensitive to grid size after the 200x200 • Not sensitive to compression value using the adaptive solver Disadvantage • By increasing C_{sh}, interface becomes sharper yet not stable for low parasitic current.
Circle in a vortex field	Fine Grid (200 x 200) • $U_f\% = 0.05$ • $n = 10$ • $C_{sh} = 0.5$ • $C_{compr.} = Adptive$		Advantage • Increase in accuracy regardless of compression Disadvantage • Snapping at tail non avoidable due to grid size effect.

6. Acknowledgements

The authors would also like to acknowledge the contribution of The Lloyds Register Foundation. Lloyds Register Foundation helps to protect life and property by supporting engineering-related education, public engagement and the application of research. Dr Vogiatzaki would like to acknowledge UKs Engineering and Physical Science Research Council support through the grant EP/P012744/1. Prof. Marengo and Dr. Georgoulas would like to acknowledge UKs Engineering and Physical Science Research Council support through the grant EP/P013112/1.

References

[1] G. M. Whitesides, The origins and the future of microfluidics, Nature 442 (2006) 368–373.

[2] M. Wörner, Numerical modeling of multiphase flows in microfluidics and micro process engineering: a review of methods and applications, Microfluidics and nanofluidics 12 (6) (2012) 841–886.

[3] R. A. Mahdi, H. Mohammed, K. Munisamy, N. Saeid, Review of convection heat transfer and fluid flow in porous media with nanofluid, Renewable and Sustainable Energy Reviews 41 (2015) 715–734.

[4] C. N. Baroud, F. Gallaire, R. Dangla, Dynamics of microfluidic droplets, Lab on a Chip 10 (16) (2010) 2032–2045.

[5] A. K. Yadav, J. C. de la Cal, M. J. Barandiaran, Feasibility of tubular microreactors for emulsion polymerization, Macromolecular Reaction Engineering 5 (1) (2011) 69–77.

[6] M. Andrew, H. Menke, M. J. Blunt, B. Bijeljic, The imaging of dynamic multiphase fluid flow using synchrotron-based x-ray microtomography at reservoir conditions, Transport in Porous Media 110 (1) (2015) 1–24.

[7] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, Journal of computational physics 79 (1) (1988) 12–49.

[8] J. Brackbill, D. Kothe, C. Zemach, A continuum method for modeling surface tension, Journal of Computational Physics 100 (1992) 335–354.

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

[10] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, Journal of Computational physics 114 (1) (1994) 146–159.

[11] T. Pringuey, R. S. Cant, Robust Conservative Level Set Method for 3D Mixed-Element MeshesApplication to LES of Primary Liquid-Sheet Breakup, Communications in Computational Physics 16 (02) (2014) 403–439.

[12] T. Pringuey, R. S. Cant, High order schemes on three-dimensional general polyhedral meshesApplication to the level set method, Communications in Computational Physics 12 (01) (2012) 1–41.

[13] D. L. Chopp, Computing minimal surfaces via level set curvature flow, Journal of Computational Physics 106 (1) (1993) 77–91.

[14] G. Russo, P. Smereka, A remark on computing distance functions, Journal of Computational Physics 163 (1) (2000) 51–67.

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

[16] H. G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Journal of Computational Physics 12 (1998) 620–631.

[17] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, G. Zanetti, Modelling merging and fragmentation in multiphase flows with SURFER, Journal of Computational Physics 113 (1) (1994) 134–147.

[18] O. Ubbink, R. Issa, A method for capturing sharp fluid interfaces on arbitrary meshes, Journal of Computational Physics 153 (1) (1999) 26–50.

[19] I. Park, K. Kim, J. Kim, S. Van, A volume-of-fluid method for incompressible free surface flows, International Journal for Numerical Methods in Fluids 61 (12) (2009) 1331–1362.

[20] V. R. Gopala, B. G. van Wachem, Volume of fluid methods for immiscible-fluid and free-surface flows, Chemical Engineering Journal 141 (1) (2008) 204–221.

[21] C. Wu, D. Young, H. Wu, Simulations of multidimensional interfacial flows by an improved volume-of-fluid method, International Journal of Heat and Mass Transfer 60 (2013) 739–755.

[22] E. Aulisa, S. Manservisi, R. Scardovelli, S. Zaleski, Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry, Journal of Computational Physics 225 (2) (2007) 2301–2319.

[23] F. Denner, B. G. van Wachem, Compressive VOF method with skewness correction to capture sharp interfaces on arbitrary meshes, Journal of Computational Physics 279 (2014) 127–144.

[24] M. Sussman, E. G. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, Journal of Computational Physics 162 (2) (2000) 301–337.

[25] G. Son, N. Hur, A coupled level set and volume-of-fluid method for the buoyancy-driven motion of fluid particles, Numerical Heat Transfer: Part B: Fundamentals 42 (6) (2002) 523–542.

[26] I. Malgarinos, N. Nikolopoulos, M. Gavaises, Coupling a local adaptive grid refinement technique with an interface sharpening scheme for the simulation of two-phase flow and free-surface flows using VOF methodology, Journal of Computational Physics 300 (2015) 732–753.

[27] Ž. Tuković, H. Jasak, A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow, Computers & fluids 55 (2012) 70–84.

[28] J. Heyns, A. Malan, T. Harms, O. Oxtoby, Development of a compressive surface capturing formulation for modelling free-surface flow by using the volume-of-fluid approach, International Journal for Numerical Methods in Fluids 71 (6) (2013) 788–804.

[29] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, International Journal for Numerical Methods in Fluids 30 (6) (1999) 775–793.

[30] G. Tryggvason, R. Scardovelli, S. Zaleski, Direct numerical simulations of gas–liquid multiphase flows, Cambridge University Press, 2011.

[31] C. Bilger, M. Aboukhedr, K. Vogiatzaki, R. Cant, Evaluation of two-phase flow solvers using Level Set and Volume of Fluid methods, Journal of Computational Physics .

[32] B. Van Wachem, A.-E. Almstedt, Methods for multiphase computational fluid dynamics, Chemical Engineering Journal 96 (1) (2003) 81–98.

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

[34] J. Roenby, H. Bredmose, H. Jasak, A Computational Method for Sharp Interface Advection, arXiv preprint arXiv:1601.05392 .

[35] S. Muzaferija, M. Peric, Computation of free-surface flows using interface-tracking and interface-capturing methods, Advanced Fluid Mechanics 24 (1999) 59–100.

[36] H. Jasak, H. G. Weller, Interface-tracking capabilities of the InterGamma differencing scheme, Tech. Rep., Imperial College London, 1995.

[37] Y. Renardy, M. Renardy, PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method, Journal of Computational Physics 183 (2) (2002) 400–421.

[38] B. Leonard, The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection, Computer methods in applied mechanics and engineering 88 (1) (1991) 17–74.

[39] S. Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, Journal of Computational Physics 228 (16) (2009) 5838–5866.

[40] F. Denner, B. G. van Wachem, Numerical time-step restrictions as a result of capillary waves, Journal of Computational Physics 285 (2015) 24–40.

[41] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, International Journal for Numerical Methods in Fluids 30 (1999) 775–793.

[42] W. J. Rider, D. B. Kothe, Reconstructing volume tracking, Journal of computational physics 141 (2) (1998) 112–152.

[43] O. OpenCFD, The Open Source CFD Toolbox, User Guide, OpenCFD Ltd .

[44] R. Issa, Solution of the implicitly discretised fluid flow equations by Operator-Splitting, Journal of Computational Physics 62 (1985) 40–65.

[45] J. H. Ferziger, M. Peric, A. Leonard, Computational methods for fluid dynamics, 1997.

[46] A. Georgoulas, P. Koukouvinis, M. Gavaises, M. Marengo, Numerical investigation of quasi-static bubble growth and detachment from submerged orifices in isothermal liquid pools: The effect of varying fluid properties and gravity levels, International Journal of Multiphase Flow 74 (2015) 59 – 78, doi:http://dx.doi.org/10.1016/j.ijmultiphaseflow.2015.04.008.

[47] A. Q. Raeini, M. J. Blunt, B. Bijeljic, Modelling two-phase flow in porous media at the pore scale using the volume-of-fluid method, Journal of Computational Physics 231 (17) (2012) 5653–5668.

[48] M. Aboukhedr, M. Gavaises, A. Georgoulas, M. Marengo, K. Vogiatzaki, Numerical Investigation of Droplet Spreading on Porous and Non-porous surfaces, 27th Annual Conference on Liquid Atomization and Spray Systems .

[49] J. D. Anderson, J. Wendt, Computational fluid dynamics, vol. 206, Springer, 1995.

[50] H. Weller, Derivation, modelling and solution of the conditionally averaged two-phase flow equations, Nabla Ltd, No Technical Report TR/HGW/02 .

[51] S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, Journal of computational physics 31 (3) (1979) 335–362.

[52] H. Jasak, H. Weller, Interface tracking capabilities of the inter-gamma differencing scheme, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine .

[53] M. Williams, D. Kothe, E. Puckett, Accuracy and convergence of continuum surface tension models, Fluid Dynamics at Interfaces, Cambridge University Press, Cambridge (1998) 294–305.

[54] R. Scardovelli, S. Zaleski, Title not needed for ILASS format, Annual Review of Fluid Mechanics 31 (1999) 567–603.

[55] Y. Sato, B. Ničeno, A conservative local interface sharpening scheme for the constrained interpolation profile method, International Journal for Numerical Methods in Fluids 70 (4) (2012) 441–467.

[56] G. Černe, S. Petelin, I. Tiselj, Numerical errors of the volume-of-fluid interface tracking algorithm, International journal for numerical methods in fluids 38 (4) (2002) 329–350.

[57] S. Arias, X. Ruiz, J. Casademunt, L. Ramírez-Piscina, R. González-Cinca, Experimental study of a microchannel bubble injector for microgravity applications, Microgravity Science and Technology 21 (1-2) (2009) 107–111.

[58] L. Ménétrier-Deremble, P. Tabeling, Droplet breakup in microfluidic junctions of arbitrary angles, Physical Review E 74 (3) (2006) 035303.

[59] V. van Steijn, C. R. Kleijn, M. T. Kreutzer, Flows around confined bubbles and their importance in triggering pinch-off, Physical review letters 103 (21) (2009) 214501.