

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Rodrigo Luís Formosinho Castello Branco

**Evaluation of Interface Capturing Algorithms for
the Prediction of Gas Bubble Ascension in
Viscous Media**

Projeto de Graduação

Projeto de Graduação apresentado ao Departamento de Engenharia
Mecânica da PUC-Rio

Orientadores: Angela Ourivio Nieckele
Bruno de Barros Mendes Kassar

Rio de Janeiro

Julho de 2019

Acknowledgments

I would like to give thanks to my advisor, Professor Angela Ourivio Nieckele, for her support, friendship and kindness to me over the course of the years I have been working with her; for all the opportunities, knowledge and countless revisions of the manuscript.

To my co-advisor, Dr. Bruno Kassar, for his friendship and support throughout the making of this work, and especially during my introduction to the OpenFOAM framework and the hours spent over the codes.

To Dr. João Carneiro for all his help and valuable suggestions to this work.

To the close friends I have made in mechanical engineering, Heitor, Patrick, Arthur, Murilo, Vitor and Rosa for their support and the good memories we've made during those tiresome years.

And to my beautiful family, for their love and care, without whom I would not be half the person I am today.

Resumo

Castello Branco, Rodrigo Luís Formosinho. Kassar, Bruno de Barros Mendes. Nieckele, Angela Ourivio. **Avaliação de métodos de captura de interface para a predição numérica da ascensão de bolhas de gás em meios viscosos**. Rio de Janeiro, 2019. 70p. Projeto de Graduação - Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Com o passar dos anos, o método Volume of Fluid original passou por diversas modificações de forma a melhorar suas capacidades de prever o comportamento de escoamentos multifásicos. O presente trabalho propõe o acoplamento de um método de cálculo de curvaturas baseado em nuvens de pontos (PC-VOF) com um método de advecção geométrico baseado no conceito de iso-superfícies (isoAdvector). Um método variante do isoAdvector, que resolve uma *Reconstructed Distance Function* (isoRDF) também foi contemplado no presente trabalho, e um acoplamento do mesmo com o PC-VOF foi realizado. Os métodos foram implementados em OpenFOAM e seus desempenhos foram avaliados com casos de benchmark. Simulações de configurações com soluções analíticas foram realizadas como testes de validação. Subsequentemente, dois casos de *benchmark* 2D que envolvem a ascensão de bolhas de gás em meios viscosos foram testados. Um dos casos constitui um escoamento dominado por tensões superficiais e o outro é dominado por forças inerciais. Malhas de quadriláteros e triângulos foram utilizadas, e testes de convergência de malha foram realizados. Os métodos foram testados e comparados através do acompanhamento da variação temporal de diversas grandezas do escoamento. Os casos de benchmark destacaram a importância da precisão no cálculo de curvatura nos escoamentos dominados por tensões superficiais, uma vez que o PC-VOF original e os métodos acoplados obtiveram as soluções mais precisas para esta configuração. Os casos também demonstraram a importância de um método de advecção preciso, particularmente na configuração dominada por forças inerciais, em que o isoAdvector e isoRDF obtiveram boas soluções. Para a maioria dos casos contemplados, os métodos acoplados apresentaram as soluções mais precisas, e são recomendados como uma alternativa estável e precisa para os métodos que os compõem.

Palavras-chave:

VOF; Bolha de gás em ascensão; Algoritmo de captura de interface; OpenFoam

Abstract

Castello Branco, Rodrigo Luís Formosinho. Kassar, Bruno de Barros Mendes. Nieckele, Angela Ourivio. **Evaluation of Interface Capturing Algorithms for the Prediction of Gas Bubble Ascension in Viscous Media**. Rio de Janeiro, 2019. 70p. Projeto de Graduação - Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Throughout the years, the standard Volume of Fluid method has undergone several modifications in order to enhance its capabilities in predicting the behavior of multiphase flow. The present work proposes the coupling of a point-cloud curvature computation method (PC-VOF) with a geometric advection scheme based on the concept of isosurfaces (isoAdvector). An isoAdvector variant method that resolves a *Reconstructed Distance Function*, called isoRDF, is also contemplated. A coupling of the latter with PC-VOF is performed. The methods were implemented in OpenFOAM, and their performances were evaluated with benchmark cases. Simulations of configurations with analytical solutions were carried out as validation tests. Subsequently two 2D benchmark cases which involve the ascension of gas bubbles in viscous media were tested. One case being driven by surface tension forces and the other by inertial forces. Both quadrilateral and triangular meshes were used, and grid convergence tests were carried out. The coupled methods, as well as their original constituents were tested and compared by tracking the time variation of a set of flow quantities. The benchmark cases highlighted the importance of curvature computation in surface tension driven flows, as the original PC-VOF and the coupled solvers presented the most accurate solutions for such configuration. They have also shown the importance of an accurate advection algorithm, particularly in the inertial forces driven flows, in which the isoAdvector and isoRDF yielded good solutions. For the majority of the contemplated cases, the coupled methods presented the most accurate solutions, and they are recommended as a stable and accurate alternative for its original constituents.

Key-words:

VOF; Gas Bubble Ascension; Interface Capturing Algorithms; OpenFOAM.

List of Figures

1.1 Flow patterns for horizontal and vertical pipes (Brennen, 2003).	12
1.2 Common interface classes (Cano Lozano et al. 2015).	14
1.3 Reconstruction methods comparison (Prosperetti & Tryggvason, 2007).	17
2.1 Distribution of the α field across a generic domain.	23
3.1 Sub-set of neighboring points around P_e (Kassar et al. 2018).	31
3.2 The time evolution of the interface position, represented by the dotted line segment, through a hexagonal cell face.	34
3.3 Resulting interface topology of a standard $\alpha = 0.5$ isovalue reconstruction method (left) and the isoAdvecting approach (right).	37
4.1 Time evolution of the aspect ratio of the oscillating bubble.	40
4.2 Time evolution of circularity of the square interface.	41
4.3 Time evolution of the interfacial shape.	42
4.4 Set-up of the two-dimensional benchmark cases (Klostermann, et al. 2013).	43
4.5 Quad. and tri. meshes for the rising bubble test (Kassar, 2016).	44
4.6 Maximum Velocity Magnitude for the TC1 zero gravity case with quad. mesh.	46
4.7 Maximum Velocity Magnitude for the TC1 zero gravity case with tri. Mesh.	47
4.8 Pressure Jump for the TC1 zero gravity case with quad. mesh.	48
4.9 Pressure Jump for the TC1 zero gravity case with tri. Mesh.	49
4.10 Pressure Fields in the bubble region for the TC1 initialization case.	50
4.11 Pressure Fields in the bubble region for the TC1 initialization case.	51
4.12 Time variation of the bubble shape for the TC1 quad. mesh configuration.	52
4.13 Time variation of the bubble shape for the TC1 tri. mesh configuration.	52
4.14 Time variation of circularity for the TC1 quadrangular mesh configuration.	53
4.15 Time variation of circularity for the TC1 triangular mesh configuration.	54

4.16 Time variation of the center of mass for the TC1 quadrangular mesh configuration.	55
4.17 Time variation of the center of mass for the TC1 triangular mesh configuration.	56
4.18 Time variation of the mean rise velocity for the TC1 quadrangular mesh configuration.	57
4.19 Time variation of the mean rise velocity for the TC1 triangular mesh configuration.	58
4.20 Grid test comparison.	59
4.21 Time evolution of the bubble shape for the TC2.	61
4.22 Time variation of curvature for the TC2 quadrangular mesh configuration.	62
4.23 Time variation of curvature for the TC2 triangular mesh configuration.	63
4.24 Time variation of the center of mass for the TC2 quadrangular mesh configuration.	64
4.25 Time variation of the center of mass for the TC2 triangular mesh configuration.	64
4.26 Time variation of the mean rise velocity for the TC2 quadrangular mesh configuration.	65
4.27 Time variation of the mean rise velocity for the TC2 triangular mesh configuration.	65
4.28 Grid test comparison for the TC2 configuration.	66

List of Symbols

α	Volume Fraction
\vec{F}_{ST}	Surface Tension Force term
σ	Surface Tension
κ	Curvature
\vec{g}	Gravity acceleration
δ_s	Dirac delta
\vec{n}	Normal vector
\hat{n}	Unit normal vector
$\tilde{\alpha}$	VOF marker function
Ψ_i	Region occupied by fluid i
\vec{x}_i	Generic point in the computational domain
\forall	Volume
ρ_i	Density of fluid i
μ_i	Viscosity of fluid i
\vec{V}	Velocity vector
$\vec{\nabla} \cdot$	Divergent Operator
∇	Gradient Operator
\vec{f}	Body forces
ϕ	Generic flow quantity
\emptyset	Volumetric face flux
Δt	Time-step
f	Cell face
A	Area
Γ	Diffusion coefficient
S	Source term

\vec{S}	Face Area vector
P	Index of the current cell
\check{A}	Anti-diffusive flux
λ	Weighing factor
Q	Net flux
K	Inflows and Outflows summation
C_α	Compressibility term
p	Pressure
p_{rgh}	Modified pressure
\vec{P}_e	Face-interface intersection point
\vec{w}	Plane normal vector
W	Weingarten matrix
e_s	Start vertex of an edge
e_e	End vertex of an edge
C	Bubble Circularity
\vec{V}_{rise}	Bubble mean rise velocity
\vec{Y}_c	Bubble center of mass

Table of Contents

Acknowledgments	2
Resumo	3
Abstract	4
List of Symbols	7
Table of Contents	9
1. Introduction	10
1.1. Literature Review	13
1.2. Objectives	20
2. Mathematical Model.....	21
3. Numerical Model.....	24
3.1 InterFoam solver	25
3.2 Point Cloud curvature computation	28
3.3 isoAdvector field advection	30
3.4 isoRDF reconstruction.....	33
3.5 Coupled methods presentation	34
4. Results and Discussion.....	36
4.1 Oscillating Drop.....	36
4.2 Square Interface.....	38
4.3 Two-dimensional Benchmark	40
4.3.1 Relaxing Condition	43
4.3.2 Test Case 1 – Rising Condition.....	49
4.3.3 Test Case 2 – Rising Condition.....	58
5. Conclusion	66
Bibliography.....	67

1. Introduction

Multiphase flows are of great importance in various areas of the industry and are characterized by the presence of multiple fluids or phases, separated by interfaces, through which the flow properties are discontinuous. They are found in refrigeration, pipeline transport of oil & gas, nuclear reactors with pressurized water, etc.

The main challenge in predicting multiphase flows is the control and tracking of the phases in the domain, which may be distributed in different patterns. In pipe flows, these patterns can be distributed in different flow configurations depending in the phase's flow rates, properties, and pipeline geometric configuration. As shown in Fig. 1.1, these arrangements can be annular, stratified, slug, churn, bubbly and dispersed. The exchange rate of mass, momentum and energy are affected by the interfacial geometric distribution of the flow components and the interfacial shape (Brennen 2003).

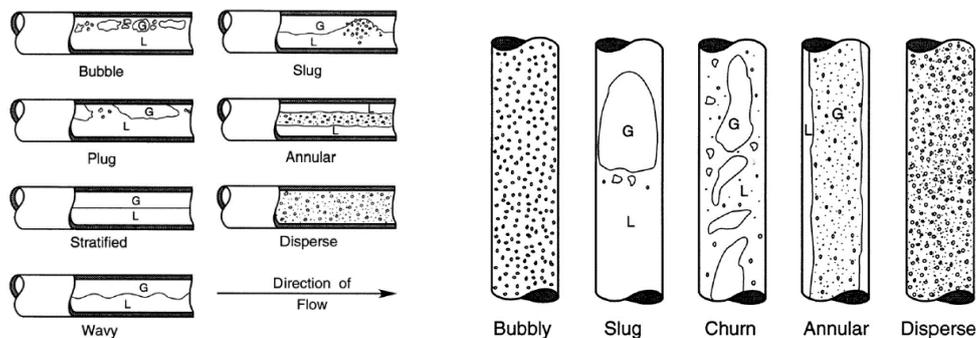


Figure 1.1 - Flow patterns for horizontal and vertical pipes (Brennen, 2003).

Bubble growth and detachment occur in various applications, such as heat exchangers, oil transport and flotation processes (Albadawi et al. 2013). Slug flow consists of a liquid phase occupying the pipe's cross-section almost entirely, flowing intermittently and separated by gas bubbles, called Taylor bubbles. The occurrence of slug flow in pipelines is often times undesirable, given the pressure jumps along the domain due to its intermittent nature. Different flow patterns are quite frequent in the industry, where transition from one pattern to another often

occurs along pipelines. The study of bubble dynamics in general or Taylor bubbles in particular has become essential for adequate pipe design.

Validation tests involving bubble dynamics are common in numerical method testing, due to their simple configuration. They also allow complex topological changes, e.g. breakup and coalescence, which present challenges for the capturing techniques (Hysing et al. 2009).

For phenomena in which the fluid interface is submitted to complex changes over time, a sophisticated numerical methodology is required. There is a wide variety of methods available in the literature that are divided into two categories: time and phase average based method, and direct interface tracking methods. A widely used method in the first category is the Two Fluid model, in which the mass, momentum and energy conservation equations are solved for each of the phases. The coupling between the phases is modelled by the interfacial terms, aided by empirical correlations. Since it requires a higher number of equations to be solved, the computational effort is usually higher in comparison to other models.

The second category, referred to as the one fluid model, solves one set of conservation equations for the entire domain, and an additional treatment is performed to keep track of the fluid interface. This is done by either solving an additional transport equation for a phase indicator, or by using marker particles to determine the interfacial points over a fixed grid (Cano-Lozano et al. 2015). The phase coupling is done by an appropriate surface tension force modelling.

Within the one fluid model, the phase indication can be done by several classes of techniques, such as Front Tracking (FT) (Univerdi & Tryggvason, 1992), Level-Set (LS) (Osher & Sethian, 1998), Marker and Cell (MAC) (Harlow & Welch, 1965) and Volume of Fluid (VOF) (Hirt & Nichols, 1981), (Prosperetti & Tryggvason 2007), (Brennen 2003).

The Front Tracking technique (Fig. 2a) employs a fixed mesh-grid for the flow field solution and a secondary moving grid of massless particles connected over the surface to keep track of the interface, which remains sharp throughout the domain. Its main drawbacks are associated with high interface deformations and topological changes. The Marker and Cell method (Fig. 2c) places marker particles in the portion of the computational domain that contains fluid, highlighting the free surface. In a similar manner to the Front Tracking method,

the particles are displaced in a Lagrangian way based on the velocity field of the fixed grid.

The Level-Set (LS) method (Fig. 2b) uses a signed distance function to identify the different flow phases. The distance function takes negative values in regions occupied by one of the fluids, and positive values in regions occupied by the other. The interface resides in the points at which the distance function is zero. The LS function ceases to act as a distance function after the first time step advection and, thus, requires re-initialization to regain its distancing properties (Albadawi et al. 2013). One of the drawbacks of this method is that it does not conserve mass.

Lastly, the VOF method (Fig. 2d) treats the interface using a scalar function α in the entire domain. This function represents the volume fraction of a particular phase chosen as a reference in each computational cell. The α function has a value of 1 in the reference phase, and 0 in the other. In cells containing the interface, α varies between 0 and 1. In regions where both phases are present, the phase properties are determined as properties of a mixture. The great attractive quality of this method for transport phenomena studies is that it guarantees mass conservation.

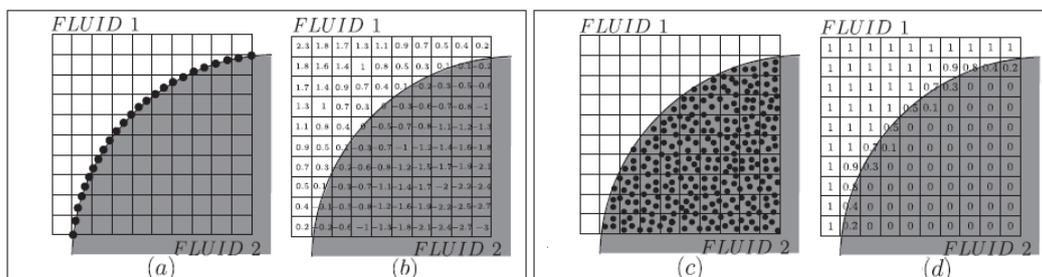


Figure 1.2 – Common interface classes (Cano-Lozano et al. 2015).

(a) Front Tracking; (b) Level Set; (c) Marker and Cell; (d) VOF.

One of the most challenging aspects of the VOF method is the evaluation of the phase transport equation. The interface can be either represented geometrically, by lines or planes, or algebraically, by functions or polynomials. The operation in geometric VOF is usually divided into two steps, an interface reconstruction and a field advection step. In algebraic VOF the transport equation is numerically approximated for α , and the volumetric fluxes are computed algebraically, therefore it does not require an interface reconstruction.

The main issue is that the use of low order schemes to discretize the transport equation leads to inaccurate solutions due to false diffusion, which gives rise to interface smearing, and high order schemes are often unstable and cause unphysical oscillations.

In order to address these issues, several methods have been developed over the years, particularly on the interface reconstruction step. Algorithms such as the Simple Line Interface Calculation (SLIC) (Noh & Woodward, 1976), Piecewise Linear Interface Calculation (PLIC) (Debar, 1974), Compressive Interface Capturing (Ubbink, 1997) and the Donor-Acceptor method (Hirt & Nichols, 1981) have been developed, some of which will be discussed in the following section.

1.1. Literature Review

The following section presents a brief discussion of the main challenges and improvements in the development of fluid interface treatment, and the role of studying ascending bubble dynamics in this particular field.

Fluid interface dynamics is of great importance to the industry and requires a careful treatment, which demands fast and robust solution methods. The VOF is one of the most widely used methods to solve fluid interfaces. However, as previously discussed, combining it with low order discretization schemes may result in smeared interfaces, and higher order schemes may generate numerical oscillations. The latter occurs because the VOF method is imprecise in the surface normal and curvature computation from direct differentiation of the volume fraction field, since it undergoes abrupt changes near the interface.

It is common to compute the surface tension in VOF by the *Continuous Surface Force* (CSF) method (Brackbill et al. 1992), in which the surface tension force can be written as $\vec{F}_{ST} = \sigma\kappa\delta_s\hat{n}$. Where σ is the surface tension, κ is the curvature, δ_s is the Dirac delta, and \hat{n} is the unit normal vector. The computation of κ and \hat{n} are crucial to an appropriate surface tension modelling. The inaccuracies in curvature computation are known to create parasitic currents and often generate unphysical pressure fields, with ripples in the interface region (Klostermann et al. 2013; Kassar, 2016).

Parasitic or spurious currents are some of the main drawbacks of the existing VOF based solvers. They can be detrimental in practical two-phase flows, leading to artificial generation of kinetic energy and heat transfer (Hardt & Wondra 2008). Besides, they can turn the solver unfeasible for surface tension driven flows. Mirjalili et al. (2017) points out that a test case assessing the relative magnitude of these currents is an essential benchmark for surface tension implementation methods.

Several works present modifications to the VOF framework in order to mitigate some of its main issues. Within the geometric approach, Noh & Woodward (1976) introduced one of the earliest interface reconstruction methods for a 2D domain, named *Simple Line Interface Calculation* (SLIC), in which the interface is approximated by straight lines in the x and y directions, where the line parallel to the x axis is used for the y direction advection, and vice e versa. Hirt & Nichols (1981) proposed a modification to the SLIC method, where instead of two straight lines parallel to each axis, a single line parallel to one of the axis is used for the advection in both directions. In order to select which line to use, the normal vector to the interface is determined, and the direction to which it is most closely aligned would be the one used.

Rider & Kothe (1998) proposed a *Piecewise Linear Interface Calculation* (PLIC) method as an improvement to the Hirt & Nichols algorithm, where the interface normals are determined, and the straight lines would be perpendicular to these normal vectors. In 3D, the straight lines would be naturally substituted by planes. Its algorithm consists of two steps: the aforementioned interface reconstruction using the interface normals as reference, and a geometrical computation of the volume fluxes across the cell faces (Rider & Kothe, 1998; Kassab, 2016). Figure 1.3 shows a comparison of the previously mentioned reconstruction methods.

The PLIC algorithm is still widely used in numerical solvers for multiphase flows, such as in Cummins et al. (2005) and Francois et al. (2006). It has also seen some improvements in recent years. Lopez et al. (2005) present an improvement to the PLIC scheme for fluid structures thinner than the cell size, by placing markers in the mid-points of every reconstructed cell in order to better track the interface segments of filaments. Mencinger & Zun (2011) present a PLIC based advection algorithm for moving grids and Ito et al. (2013) developed a PLIC

algorithm for unstructured meshes.

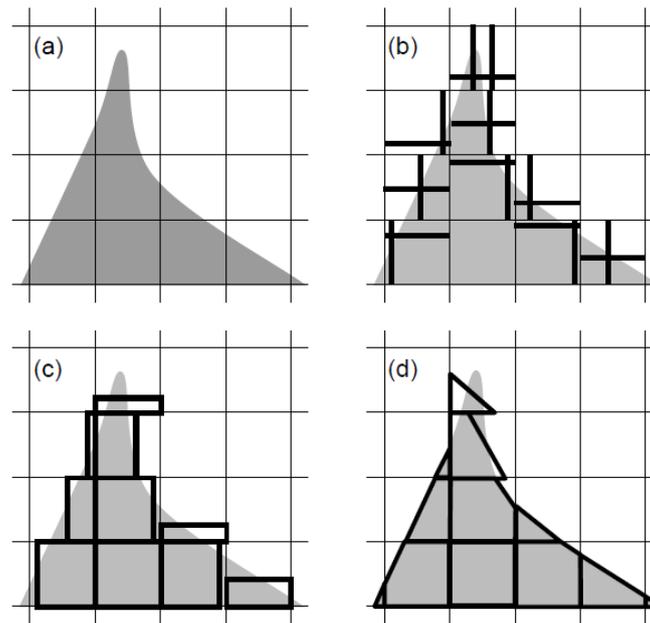


Fig 1.3 – Reconstruction methods comparison (Prosperetti & Tryggvason, 2007)

(a) The actual interface shape; (b) SLIC method; (c) Hit & Nichols; (d) PLIC.

What usually varies in the available PLIC algorithms is the interface normal computation method. Youngs (1982) has proposed an estimate to \hat{n} as the normalized gradient of the volume fraction in a computational cell. This method performs better in low resolution meshes, but it is unreliable for finer grids. Aulisa et al. (2007) developed a combined method between Youngs's and a column centered method, called mixed Youngs-Centered method (MYC), which mitigates the issue of decreased performance at higher resolution meshes. Pilliod & Puckett (2004) propose the *efficient Least-squares volume-of-fluid Interface Reconstruction Algorithm* (ELVIRA) that uses a least-squares error minimization to select an optimal normal vector.

The most common approach to obtain the interface curvature is to compute the normal vectors by spatially differentiating a smooth field, and subsequently taking the divergence of the normal vector field (Mirjalili et al. 2017), though it is quite difficult to obtain a smooth α field with sharp interface approaches. The main dilemma surrounding interface reconstruction is that a sharp approach will provide well defined interfaces, resulting in accurate calculations for the volume fluxes through the cell faces in the advection step; however, it will also narrow the interface thickness, creating rough variations of the α field across the

interface. The latter harms the normal and curvature computation performed by simple field differentiation.

A viable alternative to direct differentiation of the α field is a combination of the Level Set with the VOF method. Sussman & Puckett (2000) developed a hybrid LS and VOF method, CLSVOF, to solve two phase incompressible axisymmetric flows in 3D, aiming to model jetting apparatuses in microscale. The interface normals and curvatures are obtained from the LS's distance function, which yields superior results since the phase indicator field from the Level Set method is smoother than the VOF's α field.

Skarysz et al. (2018) presented an iterative PLIC based reconstruction method embedded in CLSVOF. Prior to the reconstruction step, the cells are divided into tetrahedra, and the volume of the truncated polyhedron can be more rapidly calculated than with the use of clipping and capping tools. It also takes advantage of the relationship between the volume fractions and interface position within a cell in its root finding algorithm.

Yang et al. (2006) presented a coupled method, called Adaptive Coupled Level-Set/Volume-of-Fluid (ACSLVOF), for unstructured triangular meshes. In this approach, the Level Set advection equation is solved with a discontinuous Galerkin method in Finite Elements.

Sussman (2003) developed a hybrid LS-VOF method, and instead of computing the curvature from the signed distance function field, a reconstruction of a Height Function directly from the volume fraction was employed. That enabled second order accurate solutions. Essentially, for a given cell, the HF method computes the so called heights by summing the volume fraction field at the neighbor cells. The resulting height values are differentiable and by employing a second-order finite difference scheme, one can obtain the curvature values. The standard Height Function method uses a fixed 7×3 stencil – 7 cells and 3 height columns – for the summation process.

Popinet (2003) developed a multiphase solver, named *Gerris Flow Solver*, that combines Adaptive Mesh Refinement (AMR) with spatial discretization in quadtree/octree for incompressible flows. Using *Gerris*, Popinet (2009) generalized the HF method, allowing second order precision on lower resolution meshes. The generalized HF method uses an adaptive stencil for the height

summation.

In particular, the use of AMR has shown significant improvement to VOF based solvers. That is due to the fact that VOF is sensitive to mesh resolution, which can be an error source for the reconstruction and advection algorithms. Low mesh refinement may cause numerical coalescence of the dispersed phase, which prevents a realistic definition of the interface when the interface separation is smaller than the size of the computational cell (Cano-Lozano et al. 2015). By concentrating mesh refinement in regions with larger gradients or abrupt changes of flow quantities, one can achieve better results with a lower computational cost.

The Convolution method (Francois et al. 2006) tackles the issue of direct differentiation of the volume fraction field by using a smooth kernel to convolute the α field. The normals and curvatures are obtained from the smooth resulting field.

Patel et al. (2017) presented a 3D hybrid method for curvature computation combining the generalized HF with the Convolution method, obtaining high accuracy at high grid resolutions and second-order convergence. Their method was validated with test cases involving stationary and oscillating droplets as well as bubble ascension in liquid columns.

Height Functions are indeed attractive in terms of accuracy, convergence and momentum conservation properties, however, their performance is weaker in lower resolution meshes, specifically where $\kappa\Delta x > 1/5$ (Popinet, 2018). This drawback in HF algorithms is ground for developing alternative interface reconstruction methods.

Renardy & Renardy (2002) proposed a surface tension parabolic reconstruction algorithm (PROST). In their work, it has been observed that the use of VOF generates spurious currents in the vicinity of the interface, and they persist in spite of mesh refinement. The PROST algorithm has overcome this problem by representing the interface by quadratic equations in the computational cells, and an optimal fit is performed in order to calculate the interface curvature value. Evrard et al. (2017) further expanded this method to unstructured meshes.

Several validation test cases for interface treatment methods involve the study of bubble behavior, particularly in recent works.

Hysing et al. (2009) proposed validation test cases for two phase flows

involving the problem of a two dimensional rising bubble in two different configurations. In their work, it is pointed out that the literature lacks quantitative data to evaluate novel numerical methods. The benchmark is composed of solutions obtained by three independent software codes: TP2D, FreeLIFE and MooNMD.

Klostermann et al. (2013) tested the standard VOF implementation in the OpenFOAM framework, the *interFoam* solver, against the validation cases proposed by Hysing et al. (2009) evaluating the influence of the surface compression scheme. They conclude that the solver is suitable for two phase flow simulations in suchlike case configurations. However, parasitic currents, faulty pressure jumps and poor grid convergence were observed in their tests.

Albadawi et al. (2013) presented the S-CLSVOF, a VOF/LS coupled method in the OpenFOAM platform. In this method, the LS is used to calculate the surface tension and the interfacial curvature. An experimental study was carried out to validate the method, where a bubble growth and detachment processes were studied. One of the obtained conclusions is that the *interFoam* standard implementation is inadequate for surface tension driven problems.

Liu & Zhao (2014) presented experimental data on bubble trajectory in stagnant water and glycerol aqueous solution, in which a three-dimensional bubble trajectory is provided. They also propose correlations combining the dimensionless numbers Weber with Eötvös and Weber with Reynolds for bubble shape.

Cano-Lozano et al. (2015) evaluated the performance of the *interFoam* and *Gerris* solvers for a case study of an ascending bubble in an initially quiescent medium. They concluded that the *interFoam* solution presented parasitic currents that remain even with mesh refinement, whereas in *Gerris* no parasitic currents were observed. It is worth noting that the *Gerris* solver employs Adaptive Mesh Refinement, whereas the standard *interFoam* implementation does not, therefore a proper comparison between both solvers should take that into account.

Kassar (2016) and Kassar et al. (2018) have proposed an approach for interface capturing with a *Point-Cloud* method, inspired by Computer Graphics techniques used in 3D scanning applications. The interface is represented by a cloud of points, and the curvatures and normal vectors are extracted from the cloud and interpolated to the computational grid. The method was implemented

in the OpenFOAM platform and has shown significant improvements to the basic *interFoam* implementation. The solvers were evaluated with the proposed validation tests by Hysing et al. (2009) and other test cases.

In geometric VOF, there are two main classes of advection schemes available: split and unsplit methods. The split methods decompose the advection equation into a set of one-dimensional advection equations, whereas the unsplit methods perform the advection in a single step.

The first developed bounded split advection method in two dimensions is the Eulerian Implicit – Lagrangian Explicit (EI-LE) method (Tryggvason et al. 2011), which was later expanded to three dimensions by Aulisa et al. (2007) in the EILE-3D method, however, it requires six advection and reconstruction steps.

Rider & Kothe (1998) proposed one of the earliest unsplit advection schemes, within the PLIC framework. Trapezoidal flux regions were created by face center velocity vectors, which could overlap, therefore breaking volume fraction boundedness. A more recent progress in this field was achieved by Ivey & Moin (2017) that presented a conservative bounded unsplit scheme, applicable on unstructured meshes.

Recently, Roenby et al. (2016) have proposed an advection numerical method, isoAdvect, that reconstructs the interface by exploring an iso-surface concept, and calculates the total transported volume of fluid through a face by the evolution of the wetted area of a reference phase within a time step. In their work, the isoAdvect method was implemented in the OpenFOAM platform as an independent solver, called *interFlow*, and was compared to different algebraic advection methods, showing superior results.

Scheufler & Roenby (2019) further expanded on the isoAdvect method by employing an interface reconstruction scheme based on the calculation of a reconstructed distance function (RDF). The RDF method was devised by Cummins et al. (2005) and involves the construction of a smooth distance function from the PLIC reconstructed interface in order to perform the curvature computation by numerical differentiation. The combination of isoAdvect and RDF yielded second order accuracy on interface normal calculations.

Su et al. (2018) evaluated the *interFoam* and *interFlow* solvers against experimental data of motions of single Taylor bubbles in vertical pipes. They observed that the simulated velocities did not agree with the experimental results,

however, they assess that the *interFlow* solver seems to be a promising tool and requires further studies.

On the algebraic VOF class, several methods of the α field advection have been developed, such as the Compressive Interface Capturing for Arbitrary Meshes (CICSAM) (Ubbink, 1997) and Flux Corrected Transport (FCT) (Boris & Book 1973).

1.2. Objectives

The present work aims at studying gas bubble ascension in viscous media, evaluating the influence of the physical parameters involved in the phenomenon. For such goal, the present work will assess the performance of some of the aforementioned methods proposed in the literature: the point cloud normal and curvature computation, PC-VOF, the isoAdvector advection scheme, along with its RDF variant, and the standard *interFoam*.

Preliminary validation tests with analytical solutions will be carried out, along with the benchmark cases proposed by Hysing et al. (2009).

This work also proposes a coupled method that attains the advantages of these different approaches, which is tested against the benchmark data and the other assessed solvers.

2. Mathematical Model

The One Fluid model resolves only one set of conservation of mass and linear momentum equations, along with an advection equation of a marker function that indicates the phase distribution of the fluids in the computational domain.

The marker function may be defined as the scalar variable $\tilde{\alpha}$ that follows the criterion below:

$$\begin{cases} \tilde{\alpha} = 1, & \text{if } \vec{x} \in \Psi_1 \\ \tilde{\alpha} = 0, & \text{if } \vec{x} \in \Psi_2 \end{cases} \quad (1.2.1)$$

where \vec{x} is the position vector on the computational domain, Ψ_1 is the region occupied by the reference phase, fluid 1, and Ψ_2 is the region occupied by fluid 2. By integrating $\tilde{\alpha}$ over the region around a specific point \vec{x}_i , one will obtain:

$$\alpha(\vec{x}_i, t) = \int_{\delta V_i} \tilde{\alpha}(\vec{x}, t) dV \quad (1.2.2)$$

In which α is the volume fraction of the reference phase at a given point and time. Figure 2.1 shows a schematic of the phase distribution of a two phase flow, where α is equal to 1 in regions occupied entirely by fluid 1, and 0 in regions entirely occupied by fluid 2 (Fig.2.1).

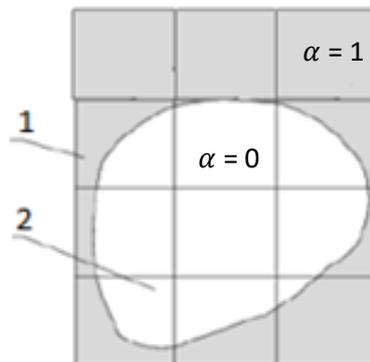


Figure 2.1 – Distribution of the α field across a generic domain.

The fluid properties can be determined as a function of α , as displayed below:

$$\rho = \rho_1 \alpha + \rho_2 (1 - \alpha) \quad (1.2.3)$$

$$\mu = \mu_1 \alpha + \mu_2 (1 - \alpha) \quad (1.2.4)$$

The present work considers an incompressible, isothermal flow, whose state can be defined by the velocity and pressure fields $v(\vec{x}, t)$ and $p(\vec{x}, t)$, and the position of the interface.

In order to determine the velocity and pressure fields, the mass conservation and linear momentum equations are employed, and are defined as:

$$\nabla \cdot \vec{V} = 0 \quad (1.2.5)$$

$$\begin{aligned} \frac{\partial(\rho \vec{V})}{\partial t} + \vec{\nabla} \cdot (\rho \vec{V} \vec{V}) = & -\vec{\nabla} p + \vec{\nabla} \cdot [\mu(\nabla \vec{V} + \nabla^T \vec{V})] + \\ & + \vec{f} + \sigma \kappa \delta(n) \hat{n} \end{aligned} \quad (1.2.6)$$

where \vec{V} is the velocity vector, p is the pressure, ρ is the density, μ is the viscosity and \vec{f} represents a body force, e.g. gravity, acting on the fluid. The term $\vec{\nabla} p$ represents the pressure gradient, and the term $[\mu(\nabla \vec{V} + \nabla^T \vec{V})]$ is the viscous stress tensor for incompressible Newtonian fluids.

The last term on the right hand side of equation 2, $\sigma \kappa \delta(n) \hat{n}$, represents the surface tension force, where σ is the surface tension, κ is twice the local mean curvature of the free surface, $\delta(n)$ is the delta function that guarantees that the term is only applied in the interface region, n is a normal coordinate to the interface, being zero at the interface, and \hat{n} is the interface unit normal vector. The normal vector and curvature terms are usually defined as:

$$\hat{n} = -\frac{\vec{\nabla} \alpha}{|\vec{\nabla} \alpha|} \quad (1.2.7)$$

$$\kappa = \vec{\nabla} \cdot \hat{n} \quad (1.2.8)$$

The surface tension is commonly modeled by the *Continuous*

Surface Force (CSF) method (Brackbill et al. 1992), where the surface tension is considered to be constant throughout the interface, and only forces that are normal to the interface are taken into account. Those considerations allow the pressure jump across the interface to be dealt as a function of the surface tension and interface curvature. Thus, the surface tension force term is added to the momentum equation to account for the jump in the surface traction in the interface region, and it constitutes the manner with which the One Fluid model treats the pressure jump boundary condition across the interface.

In order to determine the interface position across time, an advection equation is also resolved, and is defined below:

$$\frac{\partial(\alpha)}{\partial t} + \vec{\nabla} \cdot (\vec{V} \alpha) = 0 \quad (1.2.9)$$

3. Numerical Model

In a large number of CFD applications, the commercial codes utilize the Finite Volume Method (FVM) as the discretization technique, due to its conservative nature. The FVM subdivides the continuous domain into control volumes, and integrates the transport equations over all existing control volumes in the computational grid.

The general conservation equation for a generic flow quantity ϕ is given by:

$$\frac{\partial}{\partial t} \rho \phi + \nabla \cdot (\rho \vec{U} \phi) = \nabla \cdot (\Gamma \nabla \phi) + S_\phi \quad (3.1)$$

where the first term on the LHS is the time derivative and the second term is the liquid convective flux, with $\rho \vec{U}$ as the mass flux per unit area (ρ is density and \vec{U} is the velocity field). In the RHS there are the liquid diffusive and source terms, respectively. Γ represents the diffusivity coefficient.

The time derivative term is discretized by integrating it over the control volume. A discretization over a control volume with an implicit Euler scheme is written as:

$$\int_V \frac{\partial}{\partial t} \rho \phi dV = \frac{\rho_p^{i+1} \phi_p^{i+1} - \rho_p^i \phi_p^i}{\Delta t} V_p \quad (3.2)$$

in which $\phi_p^{i+1} = \phi(t + \Delta t)$ is the value taken by ϕ_p at a $t + \Delta t$ time step, whereas ϕ_p^i is the value from the previous time step. For the convective and diffusive terms, an integration over the control volume also takes place, and a subsequent transformation of the volume integral into a surface integral is done with the Gauss theorem, as shown below for the convective term:

$$\int_V \nabla \cdot (\rho \vec{U} \phi) dV = \oint_{A_f} \vec{n} \cdot (\rho \vec{U}) \phi dA = \sum_f \vec{n} \cdot (\rho \vec{U} A)_f \phi \quad (3.3)$$

where f is the face over which the integral is calculated, A is its area and \vec{n} is the face normal unit vector.

A similar process is performed for the diffusive term:

$$\int_V \nabla \cdot (\Gamma \nabla \phi) dV = \oint_A \vec{n} \cdot (\Gamma \nabla \phi)_f dA = \sum_f \vec{n} \cdot (\Gamma A \nabla \phi)_f \quad (3.4)$$

The face normal gradient can be approximated by the following scheme:

$$\vec{n} \cdot \nabla_f \phi = \frac{\phi_N - \phi_P}{|d|} \quad (3.5)$$

where d is the vector between the center of cell P and its neighboring cell N , and it is orthogonal to the face plane. Lastly, for the source term, a linearization process is performed, followed by its integration over the control volume:

$$S_\phi = S_c + S_p \phi \quad ; \quad S_p \leq 0 \quad (3.6)$$

$$\int_V S_\phi dV = (S_c + S_p \phi)_P \forall_P \quad (3.7)$$

where S_c and S_p can be dependent on ϕ .

3.1 InterFoam solver

The Open Field Operation and Manipulation (OpenFOAM) platform is a C++ open source toolbox for numerical solutions in continuum mechanics, such as computational fluid dynamics (CFD). It provides several precompiled solvers for various applications in fluid mechanics, and it is widely used in the scientific community.

Amongst the available solvers, *interFoam* handles immiscible incompressible two phase flows, and has been subject to a wide variety of tests and evaluations over the years. It is indeed a robust tool, although many authors have assessed that it is unsuitable for some applications, many of which involve surface tension driven flows, due to the recurrent appearance of spurious currents in its solution.

InterFoam is an algebraic VOF type solver, as it does not perform the surface reconstruction step. It advects the α field algebraically, using the FCT (Flux Corrected Transport) methodology.

The FCT method is a technique that guarantees that a hyperbolic

problem's solution is bounded. A demonstration of the FCT method with the transport equation of a generic flow quantity ϕ is performed in (Damián, 2013) and summarized here:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \vec{F} = 0 \quad (3.8)$$

An explicit temporal discretization of the transport equation is:

$$\frac{\phi_P^{i+1} - \phi_P^i}{\Delta t} \forall_P + \sum_f \vec{n} \cdot (\vec{F} S)_f = 0 \quad (3.9)$$

where P is the index of the current cell, and f is its face index, i is the index for the temporal steps, \forall_P is the cell volume and $\vec{S} = S \vec{n}$ is the face area vector for face f . By isolating ϕ_P^{i+1} and denoting $F^f = (\vec{F} \cdot \vec{S})_f$ as the total transport flux due to a velocity, one obtains:

$$\phi_P^{i+1} = \phi_P^i - \frac{\Delta t}{\forall_P} \left(F_{P+\frac{1}{2}}^f - F_{P-\frac{1}{2}}^f \right) = 0 \quad (3.10)$$

One can achieve boundedness of the temporal solution by either limiting the face values or face fluxes. The correction is performed by computing a flux F^L value obtained by a low order discretization scheme with guaranteed monotonic results, and an F^H obtained by a high order discretization scheme.

An anti-diffuse flux is defined as:

$$\check{A} = F^H - F^L \quad (3.11)$$

and the corrected flux, F^C , is computed as:

$$F^C = F^L + \lambda \check{A} \quad (3.12)$$

Here, λ is the weighing factor, and varies between 0 and 1. Zalesak (1979) corrects the value at time $i + 1$ firstly by a low order flux, F^L and then adding an anti-diffuse flux, \check{A} , limited by λ .

$$\phi_P^{i+1} = (\phi_P^L)^i - \frac{\Delta t}{\forall} \left(\check{A}_{P+\frac{1}{2}} \lambda_{P+\frac{1}{2}} - \check{A}_{P-\frac{1}{2}} \lambda_{P+\frac{1}{2}} \right) \quad (3.13)$$

where

$$(\phi_P^L)^i = \phi_P^i - \frac{\Delta t}{V} \left(F_{P+\frac{1}{2}}^L - F_{P-\frac{1}{2}}^L \right) \quad (3.14)$$

In this case, λ depends on the net fluxes of all faces f in cell P . The maximum and minimum net fluxes are defined as:

$$Q_P^+ = (\phi_P^{max} - \phi_P^L) \frac{\Delta t}{V} \quad (3.15)$$

$$Q_P^- = (\phi_P^L - \phi_P^{min}) \frac{\Delta t}{V} \quad (3.16)$$

and the weighing factors of \check{A} are defined below:

$$\lambda_i^\pm = \begin{cases} \min \left\{ 1, \frac{Q_i^\pm}{K_i^\pm} \right\}, & \text{if } K_i^\pm > 0, \\ 0, & \text{if } K_i^\pm < 0 \end{cases} \quad (3.17)$$

where K is the summation of all inflows and outflows for \check{A} .

InterFoam's FCT method is called Multidimensional Universal Limiter for Explicit Solution (MULES) and works similarly to the method presented by Zalesak (1979), but employs an iterative process to determine the weighing factors.

In order to decrease interface smearing, the *interFoam* implementation has an artificial compression term added to the advection equation, as shown below:

$$\frac{\partial(\alpha)}{\partial t} + \vec{\nabla} \cdot (\vec{V} \alpha) + \vec{\nabla} \cdot [\vec{V}_c \alpha (1 - \alpha)] = 0 \quad (3.18)$$

where the $\alpha(1 - \alpha)$ term is added to ensure that the compression is only performed in the interface region, and \vec{V}_c is the so called compression velocity, defined in Eq (3.19):

$$\vec{V}_c = \min[C_\alpha |\vec{V}|, \max(|\vec{V}|)] \frac{\vec{\nabla} \alpha}{|\vec{\nabla} \alpha|} \quad (3.19)$$

where C_α is the compressibility term, which can be set to zero, if one is to disregard the compressibility effect entirely, or to unity. Although MULES is less accurate than the available geometric advection schemes, it does not require surface reconstruction, which ensures mass conservation and computational efficiency (Gopala & van Wachem, 2008).

Lastly, the *interFoam* implementation of the VOF linear momentum

conservation equations is defined as:

$$\frac{\partial(\rho\vec{V})}{\partial t} + \vec{\nabla} \cdot (\rho\vec{V}\vec{V}) = \vec{\nabla} \cdot [\mu(\nabla\vec{V} + \nabla^T\vec{V})] - \vec{\nabla}p_{rgh} + \vec{F}_{ST} \quad (3.20)$$

where $\vec{\nabla}p_{rgh}$ is the gradient of the modified pressure, which incorporates the gravitational force (hydrostatic pressure), and it is defined as

$$p_{rgh} = p - \rho \vec{g} \cdot \vec{x} \quad (3.21)$$

and the \vec{F}_{ST} term represents the surface tension force, computed by the aforementioned CSF model, in which:

$$\vec{F}_{ST} = \sigma\kappa\vec{\nabla}\alpha \quad (3.22)$$

3.2 Point Cloud curvature computation

The Point Cloud VOF algorithm presented by Kassab et al. (2018) is an interface reconstruction method, and performs a superior estimation to the curvature value, $\kappa = \kappa_1 + \kappa_2$, where κ_1 and κ_2 are the principal curvatures at a given point on the interface surface.

As previously mentioned, the issue with the standard normal vector and curvature computation is that the α field undergoes abrupt changes across the interface region, resulting in an imprecise estimation of the surface tension force term. The Point Cloud method resolves this issue by applying the procedures described in this section.

At every iteration within a time step, the algorithm interpolates the α values stored in the center of each cell, to their respective cell vertexes. For every cell, it identifies the edges whose vertexes hold interpolated α values greater and lesser than 0.5, or vice e versa. Those edges are referred to as interfacial edges.

Subsequently, the method computes the coordinates of the point within the cell edge where the α value is precisely 0.5, i.e., the intersecting points between the interface surface and the computational mesh. This is performed by linear interpolation, as shown in the following equation.

$$\vec{P}_e = \vec{P}_1 + (\vec{P}_2 - \vec{P}_1) \frac{0.5 - \alpha_1}{\alpha_2 - \alpha_1} \quad (3.23)$$

where \vec{P}_e is the intersection point between the edge and the interface, \vec{P}_1 and \vec{P}_2 are the first and second vertex points and α_1 and α_2 are the α values associated with \vec{P}_1 and \vec{P}_2 , respectively.

An initial estimate of the normal vector is also performed by computing the interpolated values of the gradient of the α field in points \vec{P}_1 and \vec{P}_2 .

The next step is a geometric refinement of the normal vector by a Least-Squares Plane Fitting approach. For every interfacial edge and its interpolated point P_e , a sub-set of points in their vicinity is selected, as shown in Fig. 3.1. The sampled points belong to the interfacial edges of the neighboring cells to the point P_e .

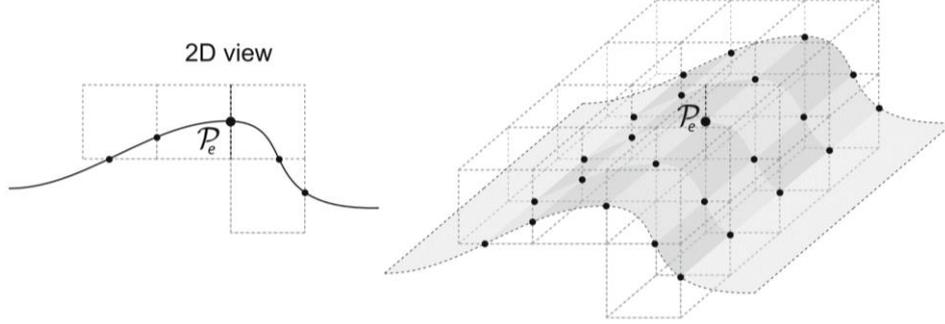


Figure 3.1. Sub-set of neighboring points around P_e (Kassar et al. 2018).

The algorithm samples two layers of the neighboring points in order to increase accuracy. This is the methodology by which the Point Cloud is generated at every reconstruction step. The normal vector is defined as the normal to the plane that best comprises the sampled cloud of points in the vicinity of P_e , by minimizing the objective function:

$$Q(\vec{w}) = \sum_{i=0}^N [\vec{w} \cdot (P_i - \langle P_i \rangle)]^2 \quad (3.24)$$

where N is the number of points in the sub-set, \vec{w} is the plane normal vector, P_i is a point from the subset and $\langle P_i \rangle$ is the centroid of the sub-set. The minimization problem is described in more detail by Kassar et al. (2018). The refined normal vector \hat{n} is taken as the resulting vector \vec{w} .

The curvature estimation for a point P_0 is performed by a normal fitting method, based on the neighboring points. The algorithm defines a local system $(\hat{u}, \hat{v}, \hat{w})$, where \hat{w} is the normal vector at P_0 , and $\hat{u} \times \hat{v} = \hat{w}$. The

local system is defined as the local frame with its origin at point P_0 .

The projection of the normal vector \hat{n}_i of the neighboring points, P_i , at the local frame is given by (n_{ui}, n_{vi}) , defined as:

$$n_u(u, v) = v_1 + w_{11}u + w_{12}v + o(u^2 + v^2) \quad (3.25)$$

$$n_v(u, v) = v_2 + w_{21}u + w_{22}v + o(u^2 + v^2) \quad (3.26)$$

where w_{11} , w_{12} , w_{21} and w_{22} are the normal vector derivatives, and v_1 and v_2 are equal to $n_{u0}(0,0)$ and $n_{v0}(0,0)$. The curvature estimation can be obtained by solving an objective function with a constraint, further explained in Kassari et al. (2018). The objective function can be simplified into two minimization problems, for the u and v components:

$$\min \sum_i^N (v_1 + w_{11}u_i + w_{12}v_i - n_{ui})^2 \quad (3.27)$$

$$\min \sum_i^N (v_2 + w_{21}u_i + w_{22}v_i - n_{vi})^2 \quad (3.28)$$

The solution of the minimization problems above provides an accurate approximation of the w_{11} , w_{12} , w_{21} and w_{22} values. The curvatures are the eigenvalues of the Weingarten curvature matrix W :

$$W = - \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \quad (3.29)$$

The curvatures are projected onto the cells, and the surface tension term on the linear momentum equation can be calculated.

3.3 isoAdvect vector field advection

Roenby et al. (2016) presented the isoAdvect scheme, which constitutes an algorithm with the two basic steps for a geometric advection method, the interface reconstruction and the advection step. The procedure is briefly described in this session.

The concept around the reconstruction step is based on the premise that the notion of the interface being defined by the points of $\alpha = 0.5$ is incorrect. They argue that the correct isovalue is the one that cuts the cell into two sub-cells with a volumetric proportion equivalent to the α value in the cell center.

The procedure performed in order to identify the interface points begins by an interpolation of the α values from the cell centers into the cell vertexes. The algorithm subsequently identifies the pairs of connected vertexes whose interpolated volume fraction f values are closest to the volume fraction of the current cell, below and above. The edge between these vertexes will contain the point with the correct isovalue. Equation (3.30) displays the described procedure.

$$x_{cut} = e_s + \frac{f - f_s}{f_e - f_s} (f_e - f_s) \quad (3.30)$$

where e_s and e_e are the start and end vertexes of an interfacial edge, f_s and f_e are their interpolated volume fractions. To precisely locate the cut point x_{cut} , another pair of points, between e_s and e_e , whose isovalues f yield α values just above and below the cell's volume fraction, are located. A geometrical evaluation of all found points result in four polynomial equations, which can be represented in a 4×4 Vandermonde matrix. Solving it by Newton's method, one can obtain the f value that yields the correct volume fraction of the cell. This, however, results in a non-continuous interface surface, because adjacent cells might not share the same isovalue.

The interface advection step tackles the challenge of estimating the volume transport of a reference phase across a face within a time step $[t, t + \Delta T]$, defined by

$$\Delta V_j(t, \Delta t) \approx \frac{\phi_j(t)}{|S_j|} \int_t^{t+\Delta t} A_j(\tau) d\tau \quad (3.31)$$

where ΔV_j is the transported volume, ϕ_j is the volumetric face flux, S_j is the normal at face j , and A_j is the submerged area in a reference fluid phase. Equation (3.31) states that integrating the time evolution within a time step of the area of a cell face submerged in a reference phase yields the volume of fluid transported across a face.

The area is calculated from the isovalue at the cell upwind of face j , and the motion of the iso-face in the time step is approximated by the velocity field in surrounding cells. This is done by finding the geometric face

center of the isoface, x_s and the unit normal vector, \hat{n} . The velocity data is then interpolated to this center point. The cell is divided into tetrahedral subcells, all containing the cell center, and the tetrahedron containing x_s is identified. The velocity value $U(x_s)$ is then interpolated to the tetrahedron vertexes, and the resulting values of U are dotted with the isoface unit normal \hat{n} .

The motion of the isosurface can be estimated by the time evolution of the intersection line between the face and the interface. Considering the vertexes of a hexagonal face shown in figure 3.2a, the time evolution can be estimated by calculating a list of times at which the face vertexes are reached by the intersection line. That estimation is defined below:

$$t_i = t + (X_i - x_s) \frac{\hat{n}_s}{U_s} \quad (3.32)$$

in which t_i varies between t and the amount of face vertexes. Figure 3.2b illustrates the time evolution by depicting the face-interface intersection at a time step t_i with the line AB, and the intersection at a further time step t_{i+1} in the line segment CD. The time varying line, $\tilde{C}\tilde{D}$, represents the interface motion. Thus, the points \tilde{C} and \tilde{D} can be found by:

$$\tilde{D}(\tau) = A + \frac{\tau - t_i}{t_{i+1} - t_i} (D - A) \quad (3.33)$$

$$\tilde{C}(\tau) = B + \frac{\tau - t_i}{t_{i+1} - t_i} (C - B) \quad (3.34)$$

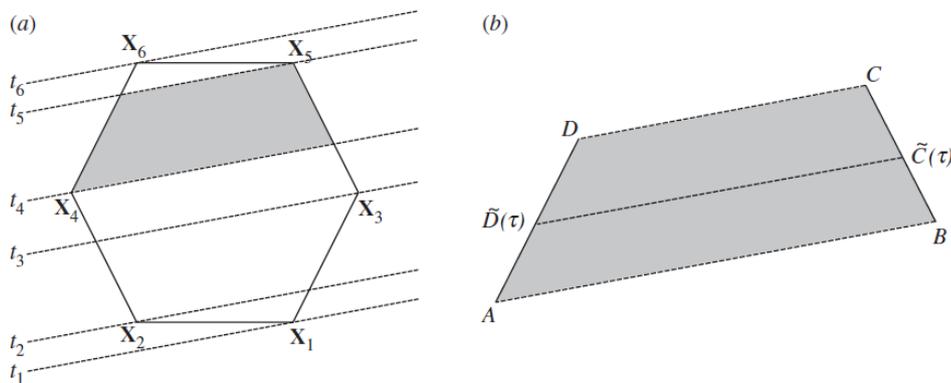


Figure 3.2 – The time evolution of the interface position, represented by the dotted line segment, through a hexagonal cell face (Roenby et al. 2016).

A list of times t' of size N is subsequently generated with values

between t and $t + \Delta t$, and the time integral in Eq. (3.35) can be divided as shown below:

$$\int_t^{t+\Delta t} A_j(\tau) d\tau = \sum_{i=1}^{N-1} \int_{t'_i}^{t'_{i+1}} A_j(\tau) d\tau \quad (3.35)$$

where $t'_i < \tau < t'_{i+1}$. The submerged area in the reference fluid can be written as:

$$A_j(\tau) = A_j(t'_i) + \frac{1}{2} \text{sign}(U) |A\tilde{C}(\tau) \times B\tilde{D}(\tau)| = \check{P}_i \tau^2 + Q_i \tau + A_j(t'_i) \quad (3.36)$$

where the polynomial coefficients \check{P}_i and Q_i can be analytically calculated from A, B, \tilde{C} and \tilde{D} . A more detailed description can be found in (Roenby et al. 2016). The contribution of the sub-intervals t'_i to t'_{i+1} can be calculated by:

$$\begin{aligned} \int_{t'_i}^{t'_{i+1}} A_j(\tau) d\tau &= \frac{1}{3} [t'_{i+1}{}^3 - t'_i{}^3] \check{P}_i + \frac{1}{2} [t'_{i+1}{}^2 - t'_i{}^2] Q_i + \\ &+ [t'_{i+1} - t'_i] A_j(t'_i) \end{aligned} \quad (3.37)$$

The solution of Eq.(3.37) gives an estimate of the amount of fluid $\Delta V_j(t, \Delta t)$ of the reference phase α that has been transported through face j .

3.4 isoRDF reconstruction

Scheufler & Roenby (2019) expanded the interface reconstruction through isovalue and volume fraction equivalence by constructing an RDF (Reconstructed Distance Function) based on the VOF data.

The RDF values are constructed by an averaging process of the distances from the cell center of a specific cell to the interfaces contained in itself and its neighboring cells. They are then stored in the cell centers. For a cell i , the RDF constructed in its cell center is defined as below:

$$D_j = \frac{\sum w_{ij} \tilde{D}_{ij}}{\sum w_{ij}} \quad (3.38)$$

where $\tilde{D}_{ij} = \hat{n}_s \cdot (x_j - x_s)$ is the distance from cell j to the interface in all

neighbor cells i , and w_{ij} is the weighing factor, defined as:

$$w_{ij} = \frac{|\hat{n}_s \cdot (x_j - x_s)|^{\check{Y}}}{|x_j - x_s|^{\check{Y}}} \quad (3.39)$$

The value of the exponent \check{Y} is also a subject of discussion. Scheufler & Roenby (2019) argue that a value of $\check{Y} = 2$ is a reasonable choice, as it is less expensive to compute.

3.5 Coupled methods presentation

In this section, a PC-VOF/isoAdvectord and a PC-VOF/isoRDF coupled methods are presented and described in detail. The goal is to combine the enhancements in normal vector and curvature computation of the PC-VOF method with the isoAdvectord and isoRDF schemes.

In the current implementation for the PC-VOF/isoAdvectord coupled method, the PC-VOF identifies the interfacial edges and extracts the normal field and curvatures based on a standard $\alpha = 0.5$ isosurface concept to be used in the surface tension force term F_{ST} . The isoAdvectord, on the other hand, reconstructs the interface based on its variable isovalue concept and performs the advection step on the reconstructed interface. Thus, there is no interaction between the methods. A similar coupling method is employed for the PC-VOF/isoRDF solver.

Earlier implementations of the coupled method had the isoAdvectord handling the reconstruction step, and the PC-VOF calculating the normals and curvatures based on the isoAdvectord reconstructed interface. The main issue with this exchange of information is that, since each interfacial cell has its own associated isosurface that cuts it into subcells with the correct volume fraction, the resulting interface is not continuous. Although this is arguably a better representation of the interface, regarding volume fraction conservation, the discontinuous surface harms the normal field and curvature estimations. Kassar et al. (2018) argues that the 0.5 fixed isovalue better represents the shape of the interface. Figure 3.3 displays a side-by-side view of the topology of a sphere generated by the fixed 0.5 isovalue method and the variable isovalue approach of isoAdvectord.

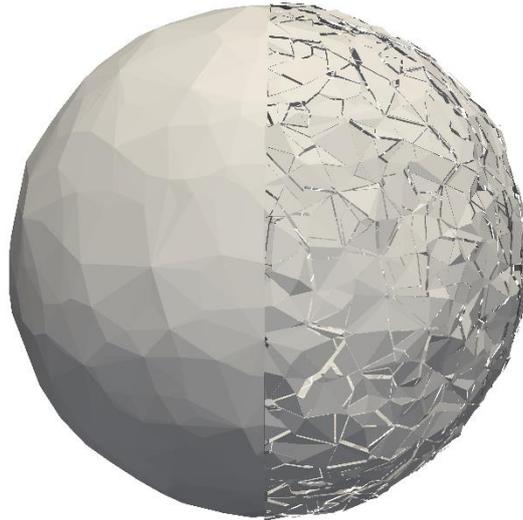


Figure 3.3 – Resulting interface topology of a standard $\alpha = 0.5$ isovalue reconstruction method (left) and the isoAdvecting approach (right).

A secondary issue is that the isoAdvecting method often generates spurious interfacial points in non-interfacial regions, due to a clipping tolerance issue. A method was created in this work aiming to identify and remove the spurious points, but it proved to be a difficult task, especially in cases where there is interface breakup.

Another iteration of the coupled method involved using the PC-VOF normal estimation to calculate the isoface unit normals for the isoAdvecting reconstructed surface, and using them to dot the interpolated velocity values $U(x_s)$ at the interfacial points x_s .

4. Results and Discussion

In order to assess the performance of the different methods contemplated in the present work, three simulation set-ups were chosen. Initially, two validation tests with analytical results were carried out. The problem of an oscillating drop embedded in a gas medium is tested for all methods, followed by a square interface problem. Lastly, benchmark cases of a bubble in a liquid column are performed, and quantitative data is generated in order to obtain a thorough evaluation of the studied flow quantities. Table 4.1 summarizes the different evaluated methods in this section.

Table 4.1 – Description of the solvers used in the present work.

Solver	Tag	Brief Description
<i>interFoam</i>	InterFoam	Standard multiphase flow solver in OpeFOAM
<i>interFlow</i>	isoAdvect	IsoAdvect geometric advection scheme
isoRDF	isoRDF	IsoAdvect coupled with an RDF scheme
PC-VOF	PC-VOF	Point Cloud curvature and normals computation
PC-VOF & isoAdvect	PC-VOF/isoAdv	PC-VOF and isoAdvect coupled method
PC-VOF & isoRDF	PC-VOF/RDF	PC-VOF and isoRDF coupled method

4.1 Oscillating Drop

The oscillating drop case consists of a two dimensional drop of kerosene immersed in air, as an approximation of a three dimensional cylinder jet. The linear theory for the oscillation frequency of the jet, ω_n , presented by Fyfe et al. (1988) as an extension to the one developed by Rayleigh (1879) is defined as:

$$\omega_n = \sqrt{\frac{\sigma(n_\omega^3 - n_\omega)}{(\rho_d + \rho_e)r_0^3}} \quad (4.1)$$

where σ is the surface tension, ρ_d is the density of the drop, ρ_e is the density

of the fluid medium, n_ω is the oscillation mode, and r_0 is the radius of the undisturbed drop. The z -axis direction of the jet is significantly greater than its radius, which makes it safe to approximate the oscillation problem into two dimensions. The flow parameters for the case with an undisturbed radius of $r_0 = 0.0125 \text{ cm}$ and oscillation mode of $n_\omega = 2$ are shown in Table 4.2.

Table 4.2: Flow parameters of the oscillating drop case.

Fluid	Density	Viscosity	Interface Tension
	$\rho(\text{kg}/\text{m}^3)$	$\mu(\text{Pa} \cdot \text{s})$	$\sigma(\text{N}/\text{m})$
Kerosene	820	0	0.03
Air	1.3	0	

The case initialization is performed under the recommendations of de Melo (1995), where the drop takes the initial shape on an ellipse with an aspect ratio of $a/b = 1.4$ given by:

$$\frac{a}{b} = \left(\frac{r_0 + r_\epsilon}{r_0 - r_\epsilon} \right) \quad (4.2)$$

where r_0 and r_ϵ are the largest and smallest axis of the ellipse. Its area A is equal to the undisturbed circular drop's area.

The air domain is defined as the rectangular area where the drop is inserted. The boundary patches of the domain edges were set to zero gradient conditions in both velocity and pressure. The simulation time was set to 3 ms .

A grid convergence test took place with quadrangular mesh spacing h of 1/32, 1/64, 1/128 and 1/256, evaluating the influence of the methods investigated in this work in their prediction of the aspect ratio a/b . The time-step was set according to the recommendations of Kassab (2016).

The standard interFoam (Fig. 4.1-a) were unable to attain a mesh independent solution with $h=1/256$, and it under predicted both frequency and amplitude. The isoAdvector (Fig. 4.1-b) method presented an unstable solution (unsmooth amplitudes), but it was able to accurately predict the frequency. However, the amplitude decreased over time, even for the finest mesh. IsoRDF (Fig. 4.1-c) under predicted the frequency, but had good

agreement for amplitude values. The PC-VOF (Fig. 4.1-d) showed far superior results in comparison to all methods.

Coupling isoAdvect and isoRDF with PC-VOF (Fig. 4.1-e, 4.1-f) showed no significant improvement for the oscillating bubble case, particularly with the PC-VOF/isoRDF, where the solution was nearly identical to the one obtained by the pure isoRDF method.

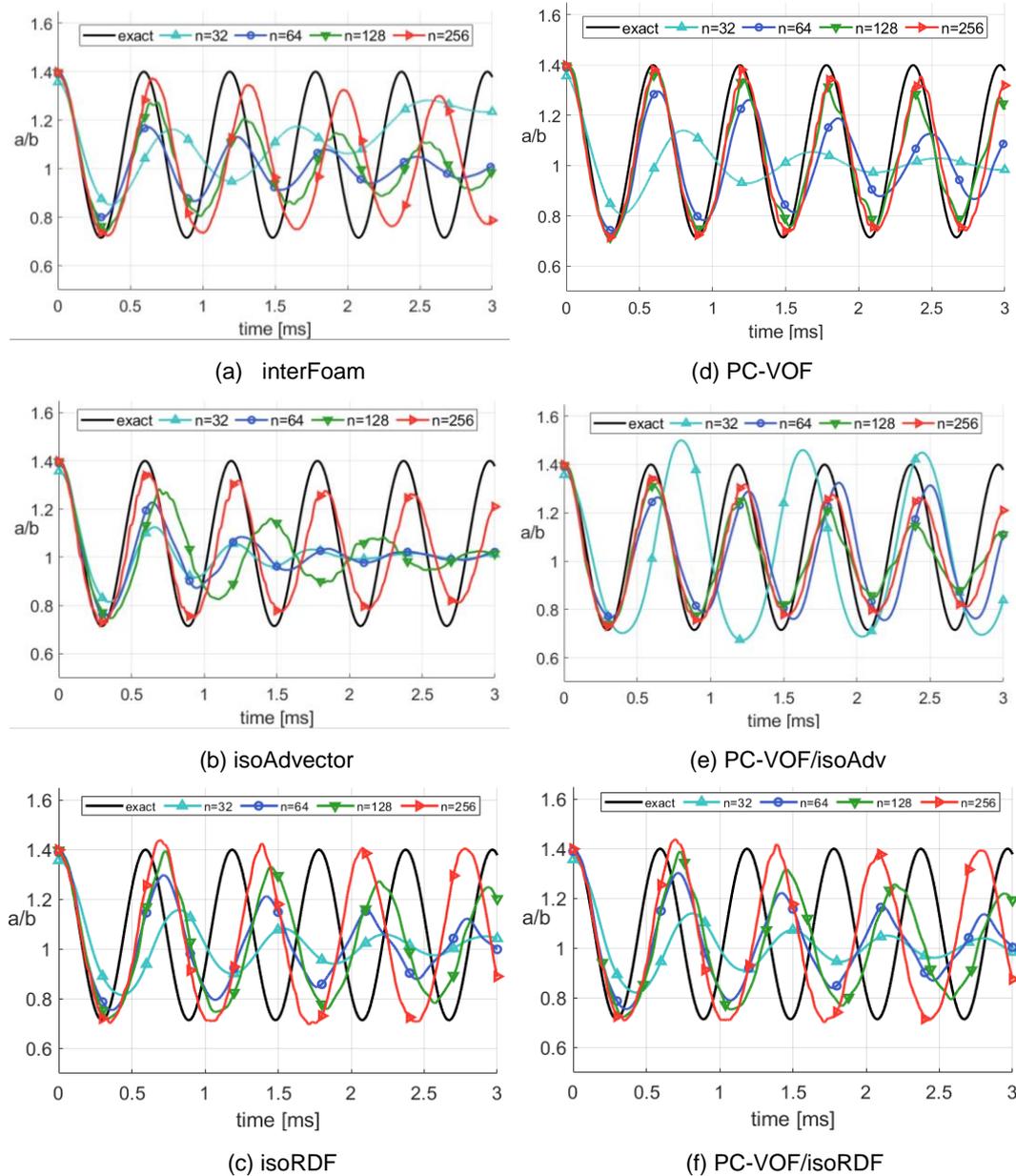


Figure 4.1 – Time evolution of the aspect ratio of the oscillating bubble.

4.2 Square Interface

The Square Interface case is a two-phase setup of a 4×4 square of

what will be referred to as fluid 2, embedded in a 10 x 10 domain of a reference fluid 1. The fluid densities and viscosities were set to unity, and a zero gravity condition was set. The goal is to track the interface as it evolves from a square into a circular shape. The case was tested for four surface tension values, $\sigma = 0.1, 1, 10$ and 100 . The simulation time was set to 20 s for the $\sigma = 1, 10$ and 100 cases and 80 s for the $\sigma = 0.1$ case, in order to ensure that steady state was achieved. A single quadrangular mesh of $h = 1/80$ spacing was selected for this case, and the time step was grounded by a maximum Courant number of $Co = 0.1$.

A useful way to analyze the main differences between the methods is to plot the time evolution of the interface's circularity. The circularity of a surface is the ratio between the circle whose area is equivalent to the surface's area, A_s , and the perimeter of the latter, P_s :

$$C_{square} = \left(\frac{A_s}{P_s} \right) \quad (4.3)$$

and it takes a value of 1 when the interfacial shape becomes a perfect circle. Figure 4.2 displays the circularity time evolution for the different case configurations.

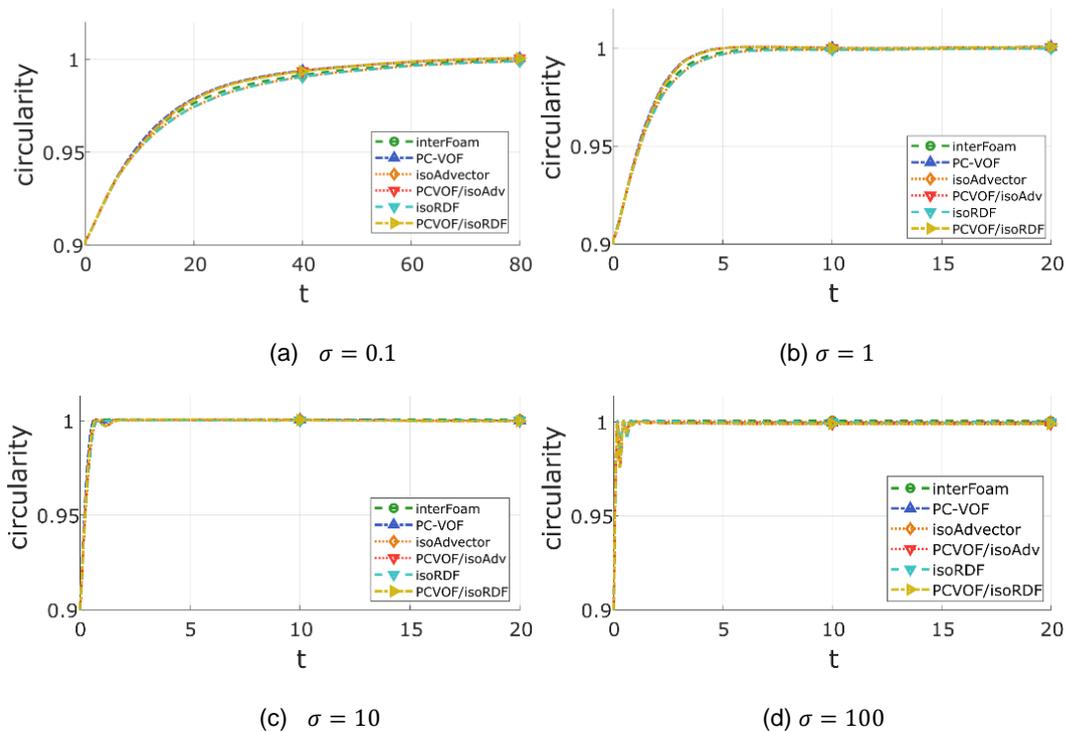


Figure 4.2 – Time evolution of circularity of the square interface.

All tested methods presented good agreement amongst themselves in regards to the deformation of the interface over time. In $\sigma = 0.1$ and $\sigma = 1$ (Fig. 4.2-a, 4.2-b) a minor variation was observed in the interFoam, isoAdvector and isoRDF cases in the transient region from the other cases. However, all cases eventually reach a circularity value of unity.

Figure 4.3 displays the evolution of the interface shape, and shows that all methods were able to accurately converge to a perfect circle, in a very short time. Although in some of the solutions the position of the circle varied, the circular shape has been preserved.

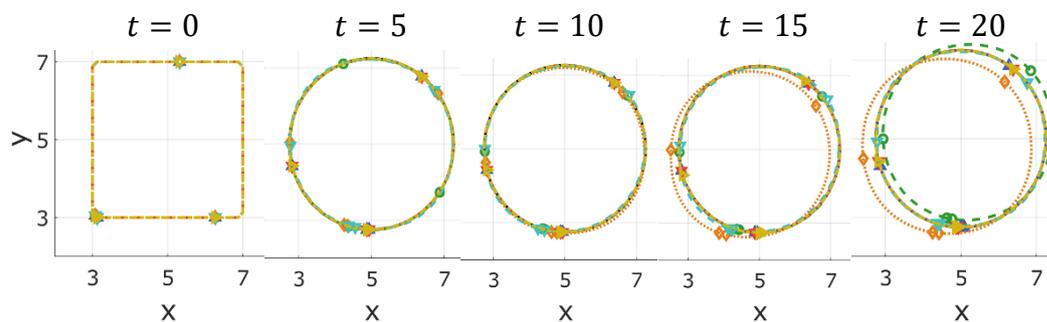


Figure 4.3 – Time evolution of the interfacial shape.

4.3 Two-dimensional Benchmark

Hysing et al. (2009) proposed a benchmark problem consisting of a two-dimensional configuration of a rising bubble in a liquid column, also providing solutions of three independent software codes. The proposed rising gas bubble case has two set-ups: Test Case 1 (TC1), which consists of a surface tension driven flow, and Test Case 2 (TC2), where the inertial forces are dominant. TC1 is particularly attractive because of its sensitivity to the different methodologies for normal and curvature calculations in the surface tension force term. In TC2, the bubble develops a skirted topology, which is also challenging to the numerical solvers.

Figure 4.4 shows the initial configuration of the test cases, in which Ψ_1 represents the liquid phase and Ψ_2 represents the gas phase. A two-dimensional rectangular domain (1×2) represents the liquid column, where the circular gas bubble is initialized, centered at $(x, y) = (0.5, 0.5)$ with an

initial radius of $r_{b0} = 0.25$. The boundary conditions at the top and bottom walls are of no slip, and free slip is set for the left and right walls.

Table 4.3 lays out the physical parameters of the test cases. The flow is governed by a set of dimensionless numbers, shown in Table 4.4: (i) Reynolds number, which is the ratio between inertial and viscous forces; (ii) Weber number, which corresponds to the ratio between inertial and surface tension forces; (iii) Eötvös number, which characterizes the ratio between body and surface tension forces; (iv) Capillary number, which corresponds to the ratio between viscous and surface tension forces, (v) Ratio of densities and (vi) ratio of viscosities. They are defined below as:

$$\begin{aligned} Re &= \frac{\rho_1 V D}{\mu_1} ; & We &= \frac{\rho_1 V^2 D}{\sigma} ; & Eo &= \frac{(\rho_1 - \rho_2) g D^2}{\sigma} ; \\ Ca &= \frac{\mu_1 V}{\sigma} = \frac{We}{Re} ; & \frac{\rho_1}{\rho_2} & ; & \frac{\mu_1}{\mu_2} & \end{aligned} \quad (4.4)$$

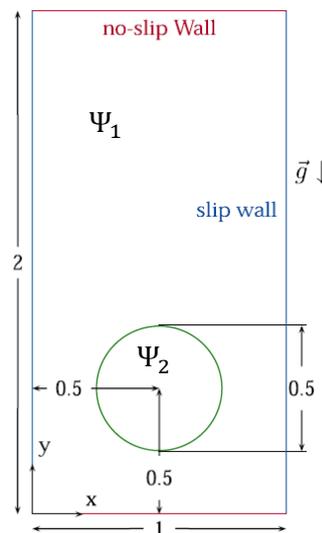


Figure 4.4 – Set-up of the two-dimensional benchmark cases (Klostermann, et al. 2013).

Table 4.3. Quantities for the benchmark configurations (Hysing et al. 2009).

TC	ρ_1	ρ_2	μ_1	μ_2	g	σ
1	1000	100	10	1	0.98	24.5
2	1000	1	10	0.1	0.98	1.96

Table 4.4. Dimensionless quantities for the benchmark cases (Hysing et al. 2009).

TC	Re	We	Eo	Ca	ρ_1 / ρ_2	μ_1 / μ_2
1	35	9.5	10	0.3	10	10
2	35	125	125	3.6	1000	100

The main idea behind the development of benchmark cases for testing numerical schemes is the use of quantitative assessments of physical properties rather than qualitative comparisons. To this end, some flow quantities were tracked: mean rise velocity \vec{V}_{rise} , center of mass \vec{Y}_c and circularity C . They are defined as:

$$\vec{V}_{rise} = \frac{\int_{\Psi_2} \vec{v} dA}{\int_{\Psi_2} dA}; \quad \vec{Y}_c = \frac{\int_{\Psi_2} \vec{x} dA}{\int_{\Psi_2} dA}; \quad C = \frac{2\pi r_{b0}}{P_b} \quad (4.5)$$

where Ψ_2 represents the region occupied by the gas bubble, \vec{V}_{rise} is the rise velocity, $2\pi r_{b0}$ is the perimeters of the initial circular bubble and P_b is the perimeter of the deformed bubble.

For both TC1 and TC2, a grid test with four mesh resolutions was selected, with mesh spacing of $h = 1/40, 1/80, 1/160$ and $1/320$. Both quadrangular and triangular meshes were employed. Figure 4.5 shows these two mesh arrangements for the largest mesh size ($h = 1/40$).

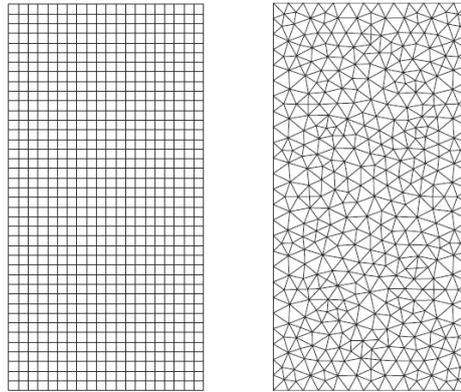


Figure 4.5 – Quad. and tri. meshes for the rising bubble test (Kassar, 2016)

Prior to simulating the rising bubble problem, an initialization test with a zero gravity condition was carried out in order to relax the Heaviside aspect of the α field profile in interfacial regions. The solution from the relaxing bubble is usually employed as initial conditions for the rising bubble simulation. This test is particularly useful to evaluate the appearance of spurious currents in the interface region. In the present work, the initialization test was only performed for the Test Case 1 configuration in quadrangular and triangular meshes, and its results are presented below.

4.3.1 Relaxing Condition

In these tests, a zero gravity condition is set to the entire domain. The viscous forces must mitigate any translational velocity ($\vec{u} = 0$) both in the gas bubble region and on the liquid domain. The initial position is expected to be maintained at $\vec{x} = (0.5, 0.5)$, and the simulation is run for 3 seconds.

The occurrence of spurious velocities, as previously mentioned, is due to improper computation of the curvature values in the CSF model, and are known to generate unphysical pressure jumps.

For the two dimensional bubble, the curvature is given by $1/r_0$, and the pressure jump is given by:

$$\Delta P = \sigma \kappa = \frac{\sigma}{r_0} \quad (4.6)$$

For TC1, the pressure jump is $\Delta P = 98 \text{ Pa}$. In order to evaluate the pressure fields of the selected numerical models, the pressure jump is computed as

$$\Delta P = \frac{\int_{\Psi_2} p dV}{\int_{\Psi_2} dV} - \frac{\int_{\Psi_1} p dV}{\int_{\Psi_1} dV} \quad (4.7)$$

Figure 4.6 depicts the time evolution of the maximum magnitude velocity field generated by the spurious currents for quadrangular meshes. By analyzing Fig. 4.6, it is possible to identify a reduction in the spurious currents by the PC-VOF algorithm (Fig. 4.6-d) in comparison to the standard interFoam (Fig. 4.6-a), although the latter is able to reduce the spurious currents over time. PC-VOF presented a significant amount of numerical oscillations for the $h = 1/320$ grid. The spurious velocities in isoAdvector (Fig. 4.6-b) have increased in comparison to interFoam, and its values were relatively constant throughout the simulation time. The maximum velocity magnitude obtained by the isoRDF (Fig. 4.6-c) solver had nearly undetectable differences when compared to the one obtained by isoAdvector. isoAdvector and isoRDF were not able to obtain mesh refinement convergence, unlike the other solvers. The PC-VOF/isoAdvector coupled method (Fig. 4.6-e) had similar results to PC-VOF along with a

more stable solution for the $1/h = 320$ mesh. Coupling isoRDF with PC-VOF (Fig. 4.6-f) also yielded similar results to the PC-VOF/isoAdvect method.

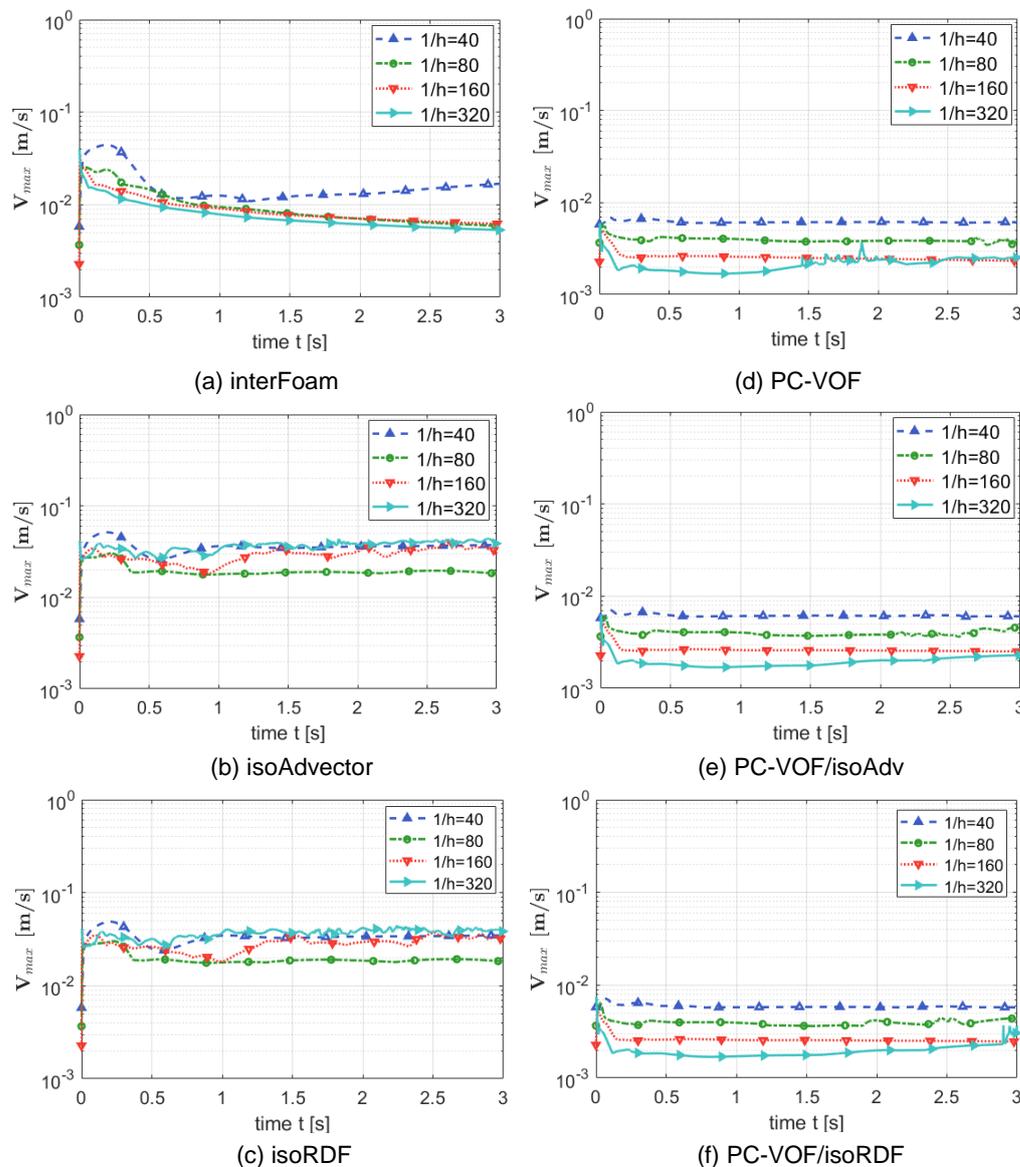


Figure 4.6 – Maximum Velocity Magnitude for the TC1 zero gravity case with quad. mesh.

The velocity fields on triangular or unstructured meshes saw an overall increase in magnitude when compared to quadrangular meshes for all methods. The solutions obtained by isoAdvect (Fig. 4.7-b) and isoRDF (Fig. 4.7-c) have also presented a certain amount of numerical oscillations. The original PC-VOF (Fig. 4.7-d) achieved promising results, with its maximum velocity converging to values with orders of magnitude of 10^{-2} m/s. However, the coupled methods (Fig. 4.7-e, 4.7-f) obtained a more stable time evolution with similar orders of magnitude.

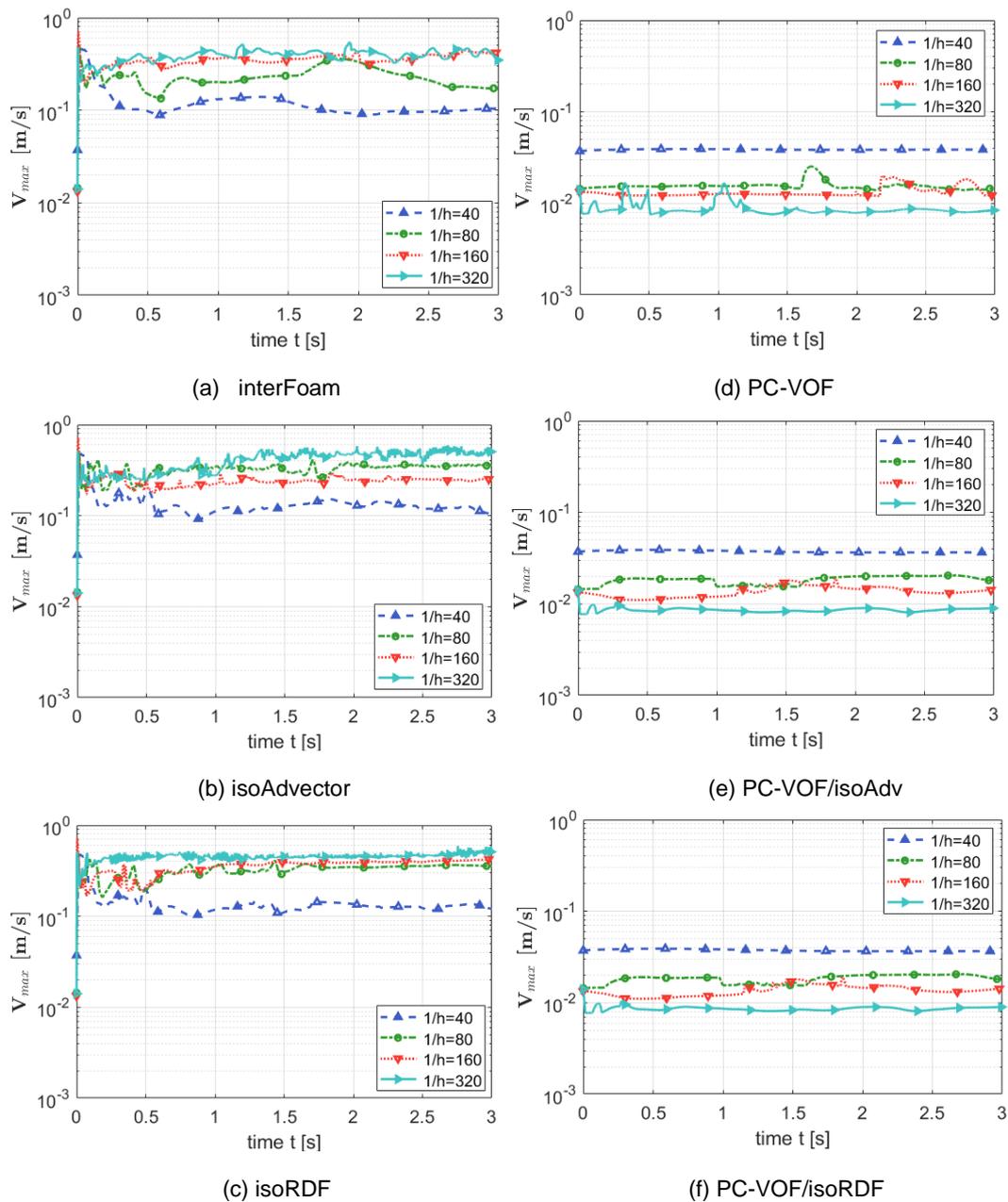


Figure 4.7 – Maximum Velocity Magnitude for the TC1 zero gravity case with tri. Mesh.

Figure 4.8 displays the pressure jump ΔP values over time. The interFoam (Fig. 4.8-a) solution had a higher error against the theoretical data, but presented a rather stable solution as well. The pressure jump prediction by the isoAdvector (Fig. 4.8-b) solver was unstable, did not achieve mesh convergence and reached significantly lower values when compared to the other tested solvers. The isoRDF presented more stable results (Fig. 4.8-c) than the original IsoAdvector. All solutions obtained with the PC-VOF method, alone (Fig. 4.8-d) or coupled (PC-VOF/isoAdvector in Fig. 4.8-e) and (PC-VOF/isoRDF in Fig. 4.8-e) are significantly better, in agreement with the

theoretical value, proved to be stable throughout the simulation time, and showed convergence with mesh refinement.

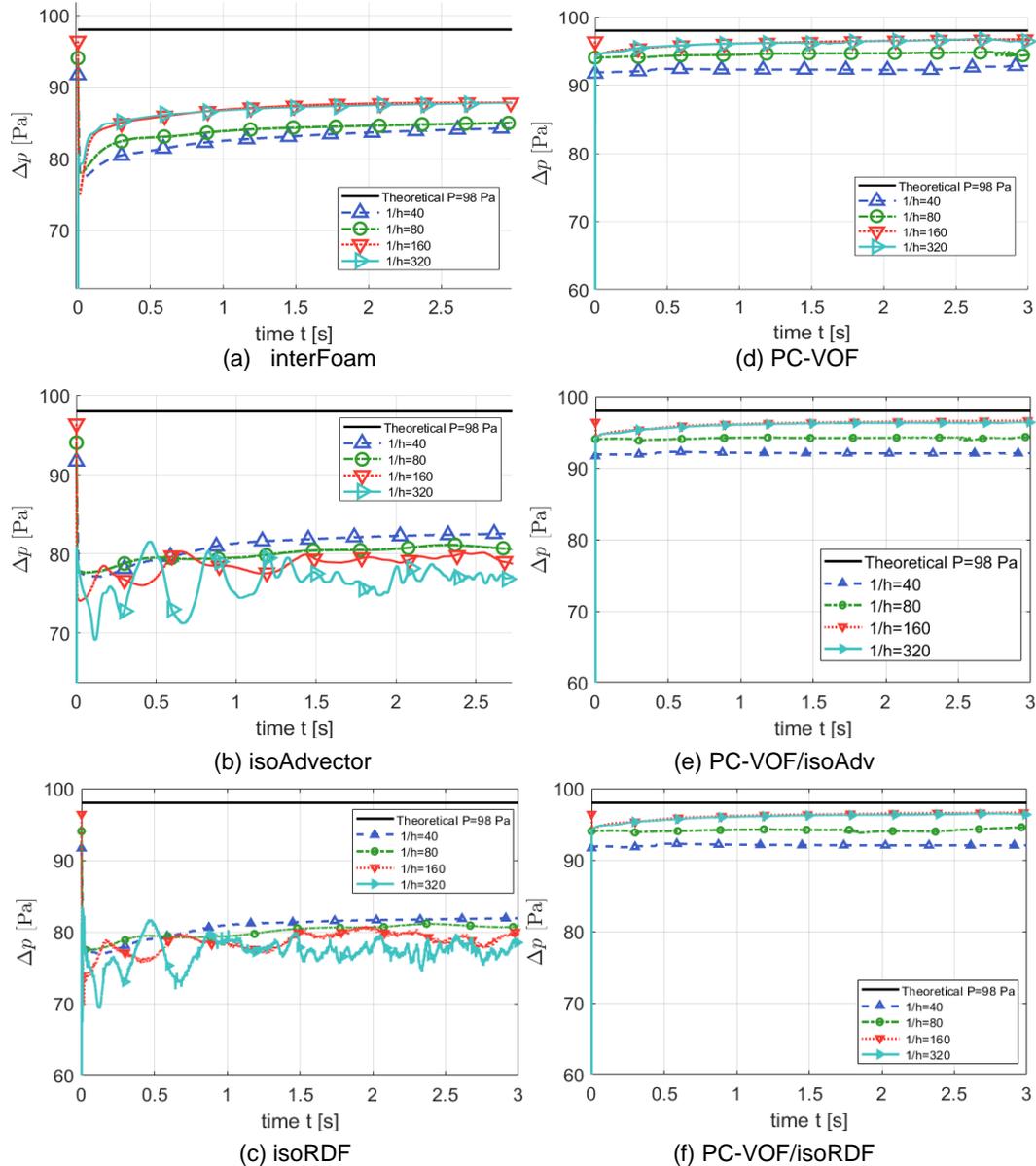


Figure 4.8 – Pressure Jump for the TC1 zero gravity case with quad. mesh.

For the unstructured grids (Fig. 4.9), the results highlighted the lacking performance of interFoam (Fig. 4.9-a), isoAdvector (Fig. 4.9-b) and isoRDF (Fig. 4.9-c), which prove to be unfit for surface tension driven flows. On the other hand, the PC-VOF (Fig. 4.9-d) and coupled methods (Fig. 4.9-e, 4.9-f) obtained a significantly more stable solution, although the prediction of the pressure jump value has deteriorated.

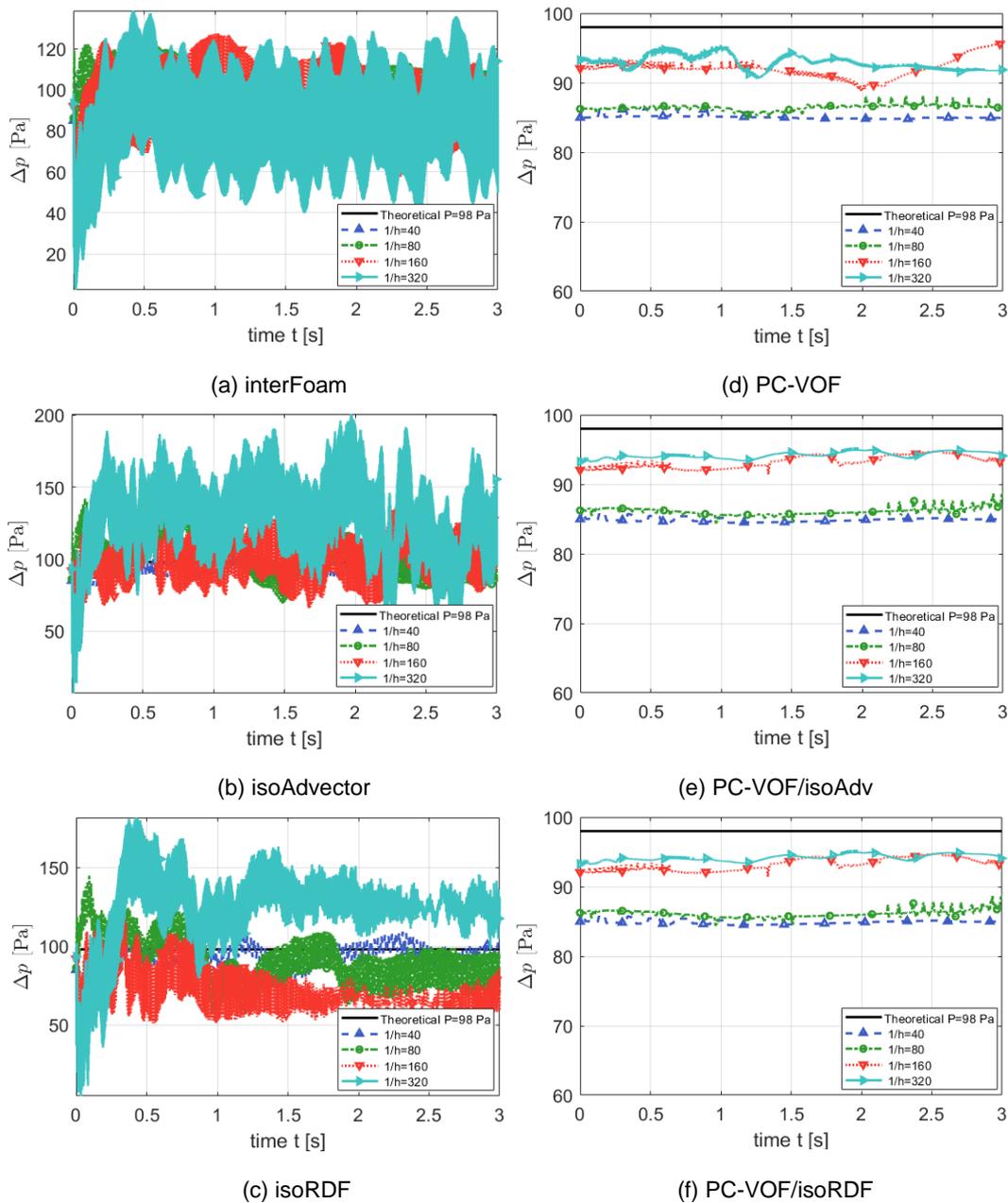


Figure 4.9 – Maximum Velocity Magnitude for the TC1 zero gravity case with tri. Mesh.

A useful way to evaluate the solution for pressure is to plot the pressure fields in the last time step of the simulations, as depicted in Fig. 4.10. The pressure field is normalized by the pressure jump across the interface $\Delta P = 2\sigma/r_{b0}$, resulting in $p^* = pr_{b0}/2\sigma$. The p^* field ranges from 0 in the liquid region to 1 in the gas bubble region. The pressure field of the interFoam solver (Fig. 4.10-a) showed a significant amount of ripples in the interface region, but overall a rather stable field was achieved. The fields obtained by the isoAdvector (Fig. 4.10-b) and isoRDF (Fig. 4.10-c) solvers were unstable and presented a large amount of ripples. PC-VOF (Fig. 4.10-

d) obtained a pressure field with virtually no ripples. The coupled methods obtained identical solutions to the pure PC-VOF method. On unstructured meshes, the pressure fields generated by interFoam (Fig. 4.11-a), isoAdvector (Fig. 4.11-b) and isoRDF (Fig. 4.11-c) were significantly unstable, with several pressure spikes in an almost unrecognizable cylindrical shape. The PC-VOF based methods obtained pressure fields almost identical to the quadrangular meshes.

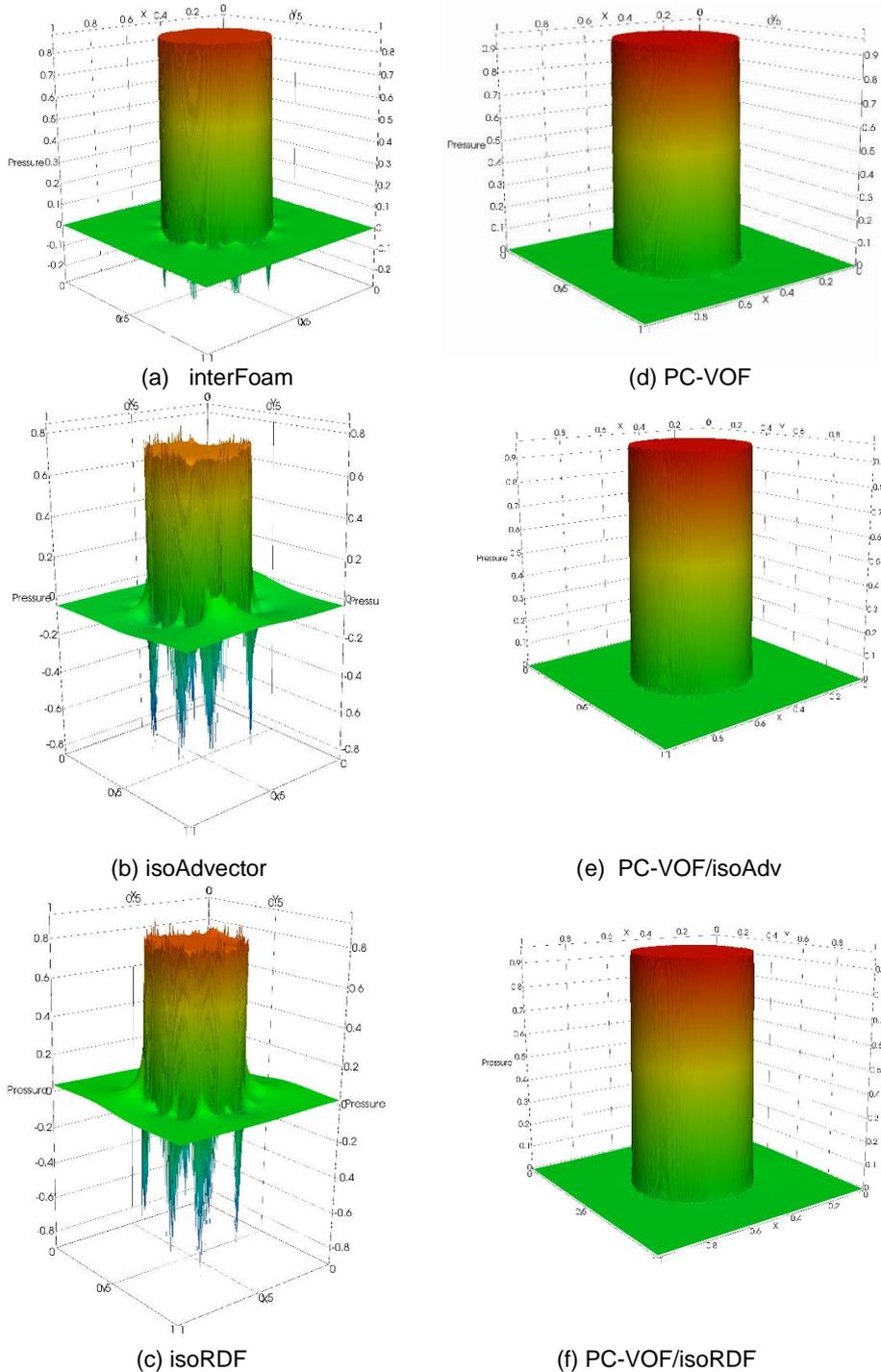


Figure 4.10 – Pressure Fields in the bubble region for the TC1 initialization case.

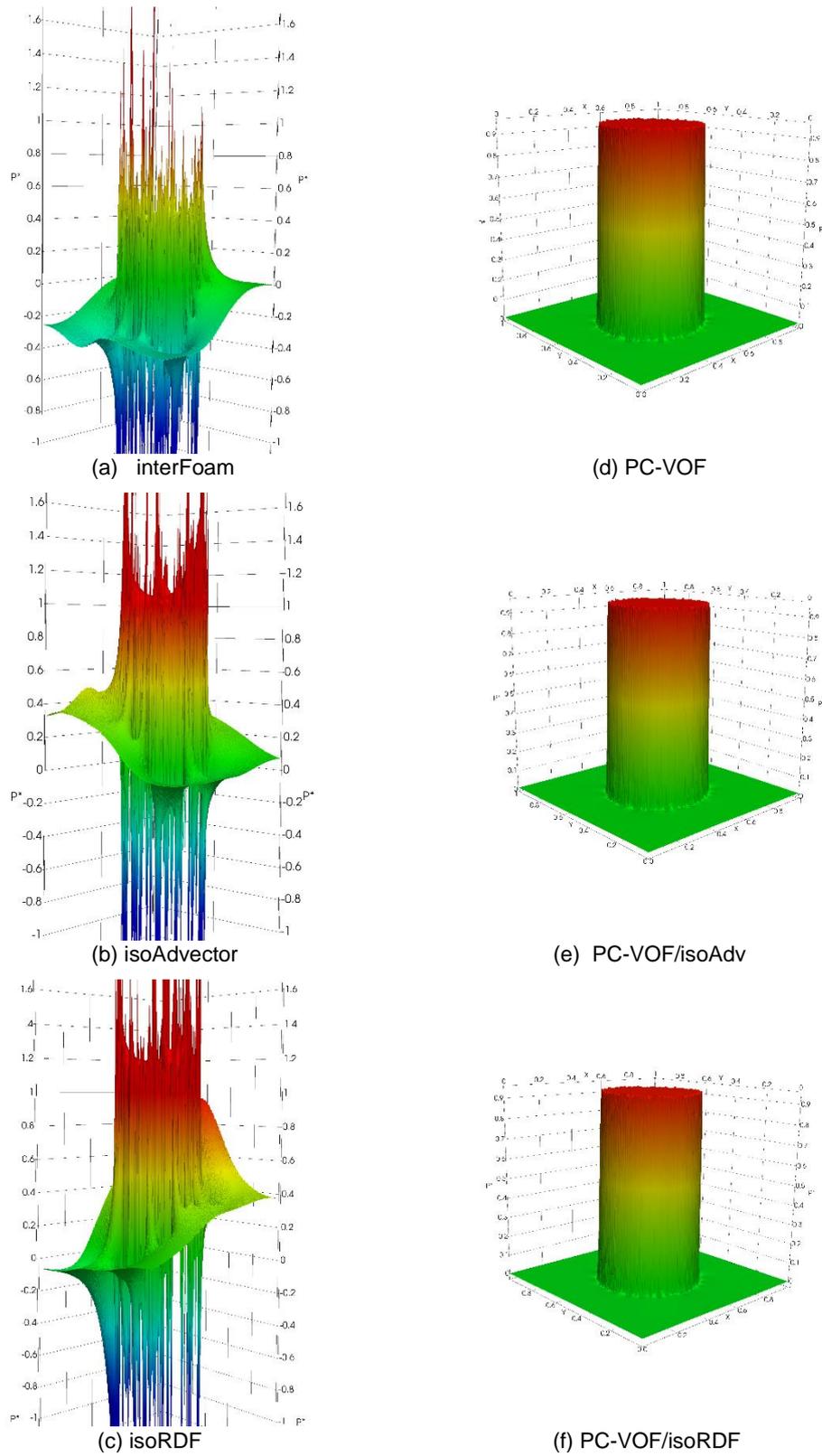


Figure 4.11 – Pressure Fields in the bubble region for the TC1 initialization case.

4.3.2 Test Case 1 – Rising Condition

Following the zero gravity condition case, the rising bubble simulation

with the TC1 configuration took place. Hysing et al. (2009) presented solutions for the rising bubble tests from three different finite element solvers, TP2D, FreeLIFE and MooNMD. The solutions obtained by the methods contemplated in this work were compared to these results.

For the two dimensional TC1 case, the simulation was run for 3s and results for mean rise velocity, center of mass and circularity are laid out below for quadrangular and triangular meshes.

During the bubble ascension in the liquid column, its interface topology undergoes a significant amount of deformation, gaining a concave shape in its lower region as it widens and loses its circular form (Figs. 4.12 and 4.13).

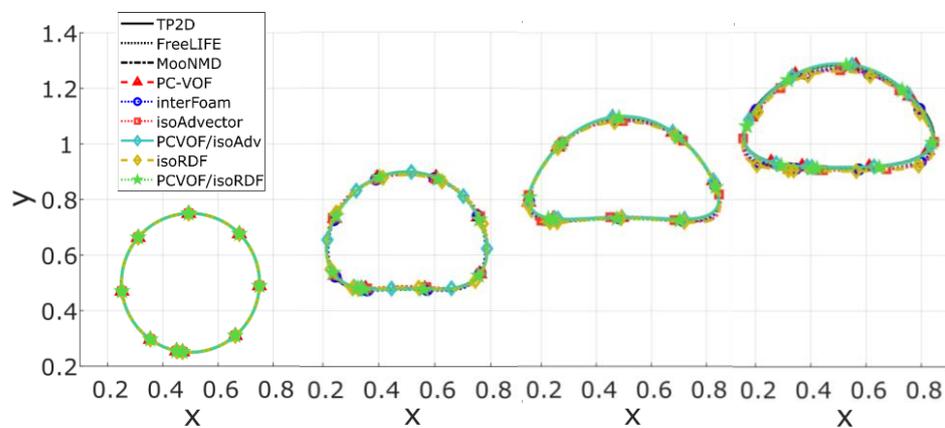


Figure 4.12 –Time variation of the bubble shape for the TC1 quad. mesh configuration.

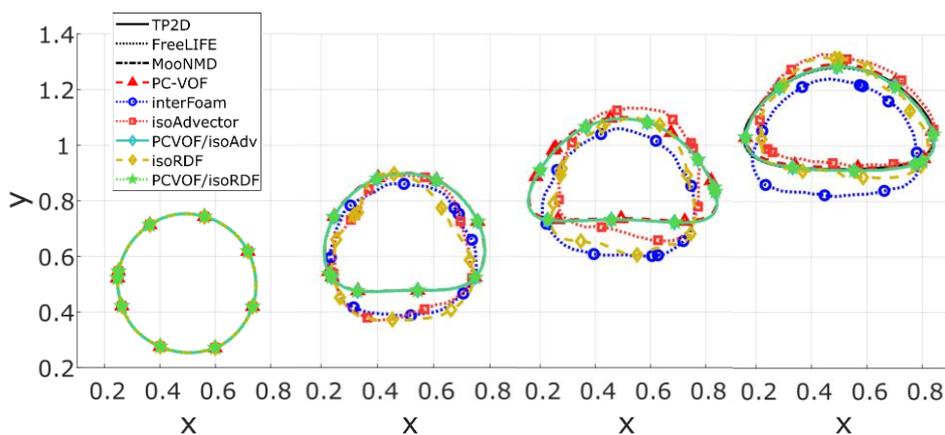


Figure 4.13 –Time variation of the bubble shape for the TC1 tri. mesh configuration.

This pattern is well predicted in all tested methods for quadrangular meshes, as shown in Fig. 4.12 for mesh resolution of $h = 1/160$. For triangular grids, the solvers that employ the PC-VOF curvature computation managed to accurately predict the bubble shape, however, interFoam,

isoAdvector and isoRDF were unable to do so, as shown in figure 4.13 for $h = 1/160$ grid spacing. Figure 4.14 lays out the grid convergence test for circularity. The tested methods have all reached convergence with mesh refinement, however, the interFoam (Fig. 4.14-a), isoAdvector (Fig. 4.14-b) and isoRDF (Fig. 4.14-c) solutions did not agree with benchmark data, whereas all models that employed the PC-VOF method, alone or coupled (Fig. 4.14-d, 4.14-e and 4.14-f), presented good agreement with the benchmark cases.

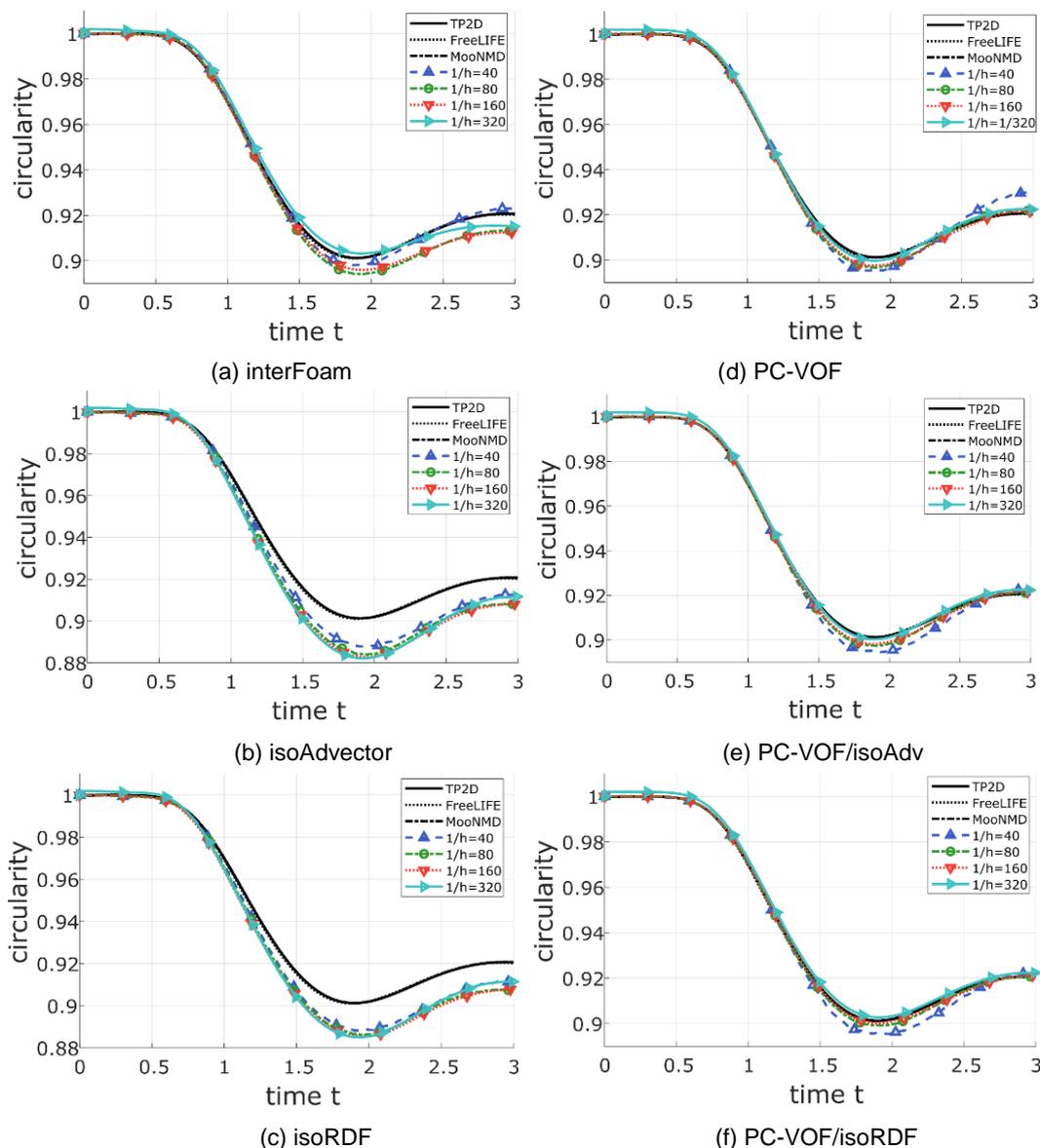


Figure 4.14 –Time variation of circularity for the TC1 quadrangular mesh configuration.

Figure 4.15 below shows the grid test results for triangular meshes. The interFoam (Fig. 4.15-a) and isoAdvector (Fig. 4.15-b) solvers did not

obtain mesh convergence, and isoRDF's $h = 1/320$ solution (Fig. 4.15-c) did not converge. The PC-VOF based methods have all followed a similar pattern. The coupled methods (Fig. 4.15-e, 4.15-f), in particular, have obtained a more accurate solution than the pure PC-VOF method (Fig. 4.15-d) for coarser mesh resolutions, however, the difference is quite subtle.

The accurate solutions obtained by the original PC-VOF and the coupled methods highlight the importance of curvature computation to the interface topology prediction, as opposed to a more enhanced advection algorithm, as in the isoAdvectord case.

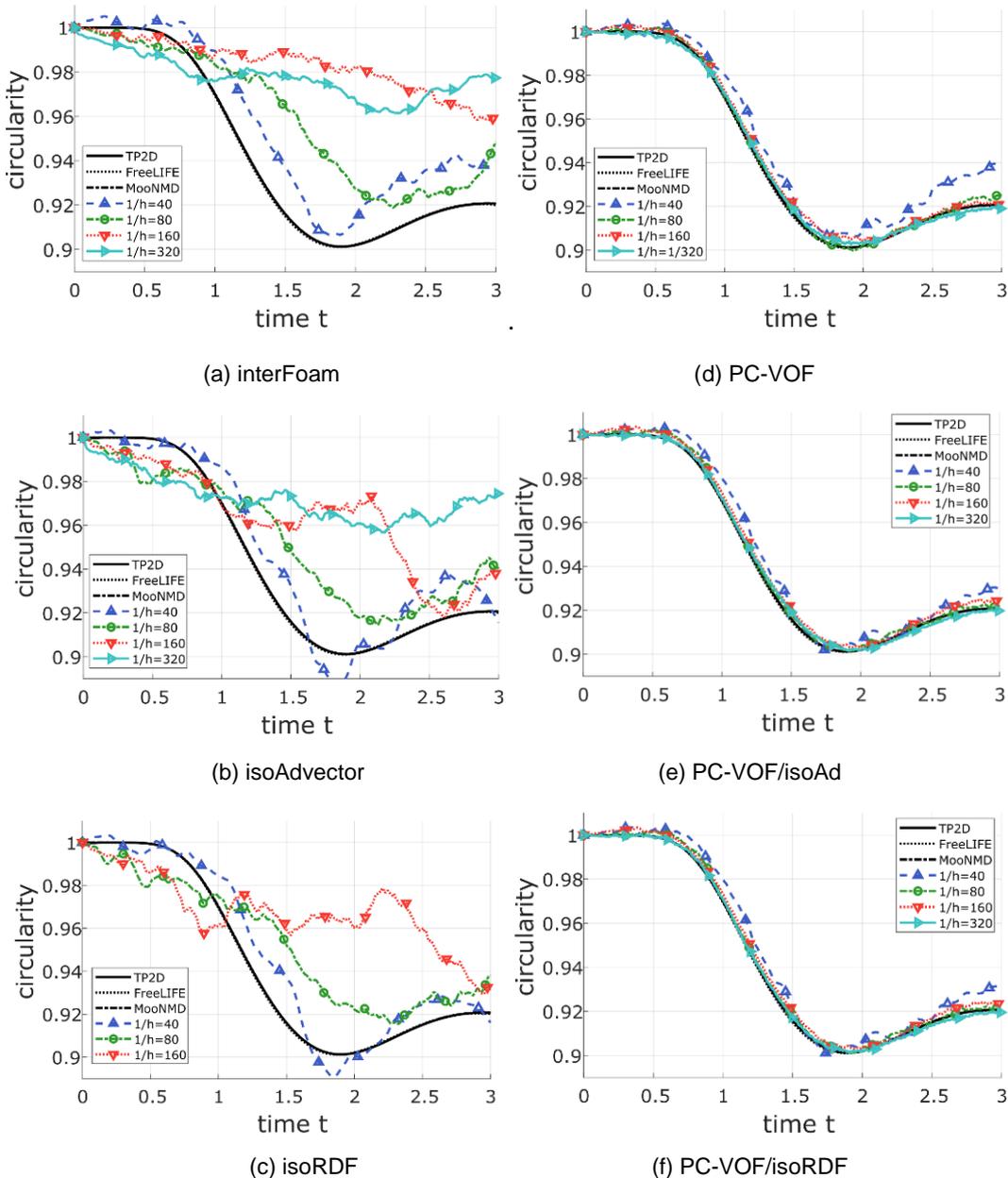


Figure 4.15 – Time variation of circularity for the TC1 triangular mesh configuration.

The time evolution of the center of mass results for structured meshes are displayed in Figure 4.16. All methods obtained good agreement to benchmark data, with interFoam, isoAdvector and isoRDF showing minor deviations at later time steps (approximately $t > 1s$). An analysis of the triangular mesh results (Fig. 4.17) shows that the interFoam, isoAdvector and isoRDF did not converge with mesh refinement, and, as previously mentioned, the isoRDF $h = 1/320$ solution did not converge. The PC-VOF solvers had virtually identical solutions for quadrangular and triangular meshes.

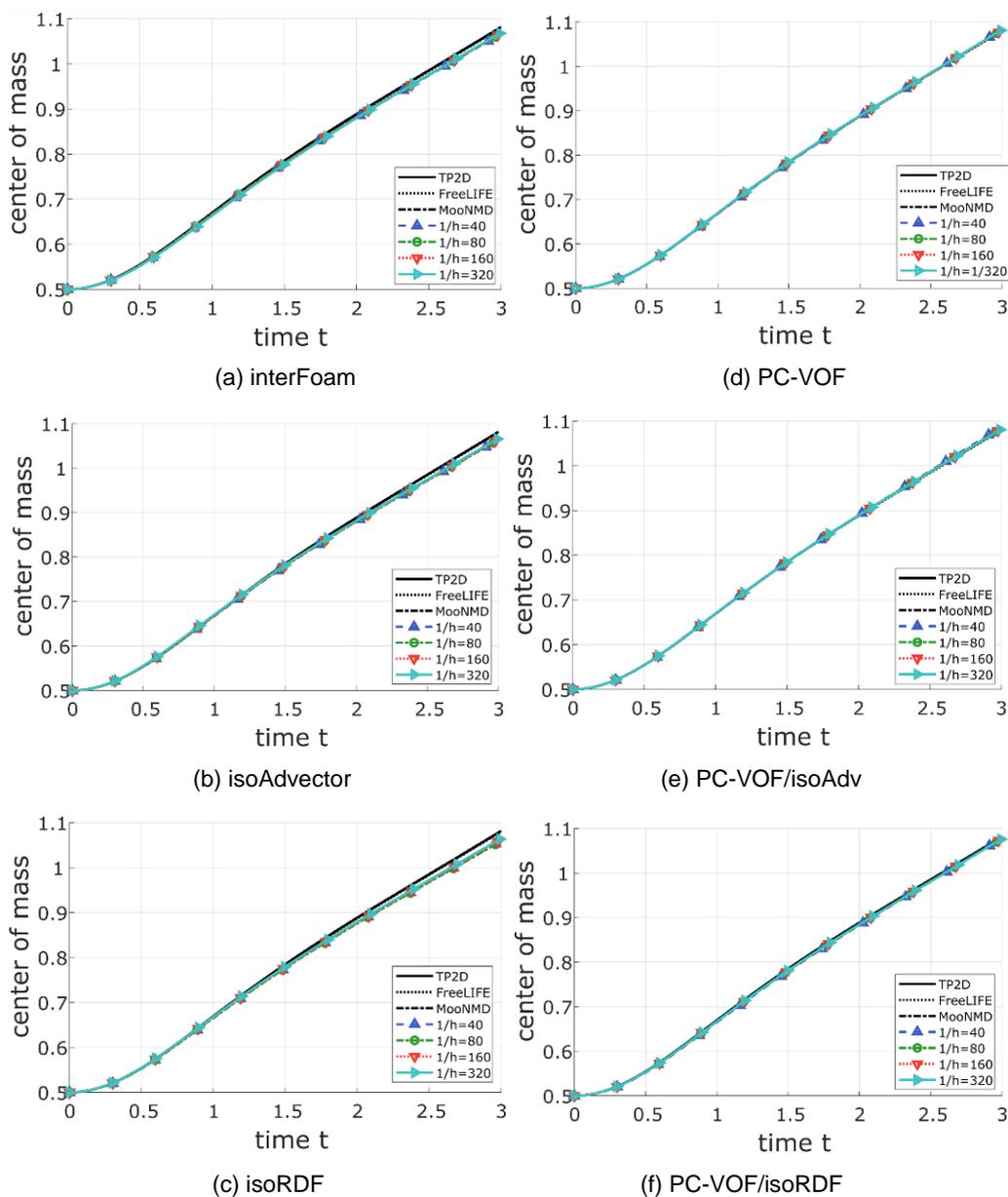


Figure 4.16 –Time variation of the center of mass for the TC1 quadrangular mesh configuration.

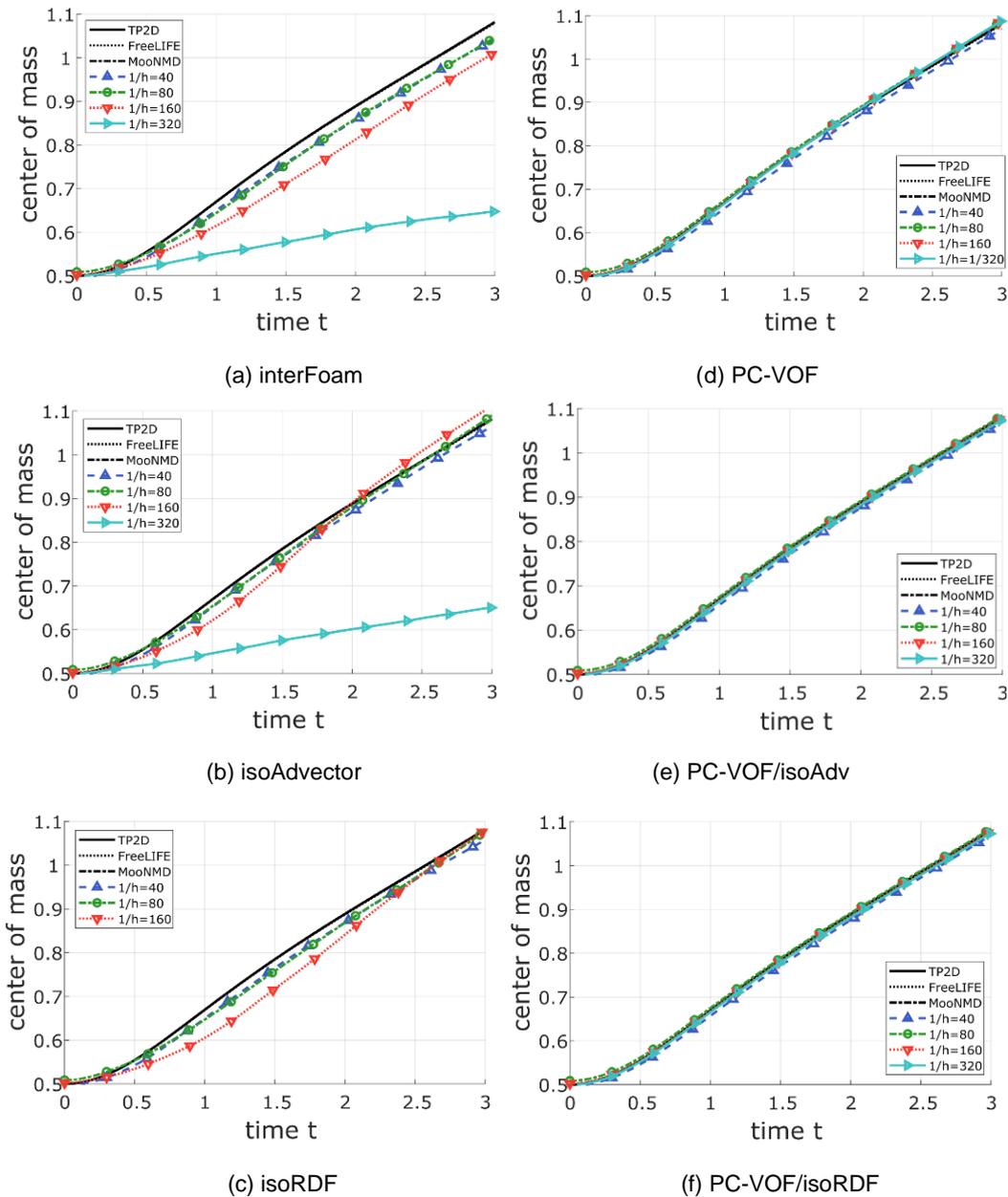


Figure 4.17 –Time variation of the center of mass for the TC1 triangular mesh configuration.

For the time variation of rise velocity (Fig 4.18) the PC-VOF and the PC-VOF/isoAdv obtained identical solutions to the benchmark data. The PC-VOF/isoRDF had minor deviations from the benchmark solutions once the rise velocity began to stabilize at $V_{rise} > 0.2 \text{ m/s}$.

The isoAdvectord and isoRDF cases were able to predict the slope region accurately, and their solutions suffered small deviations at $t > 0.5$. Overall, they under predicted the rise velocity by a small account.

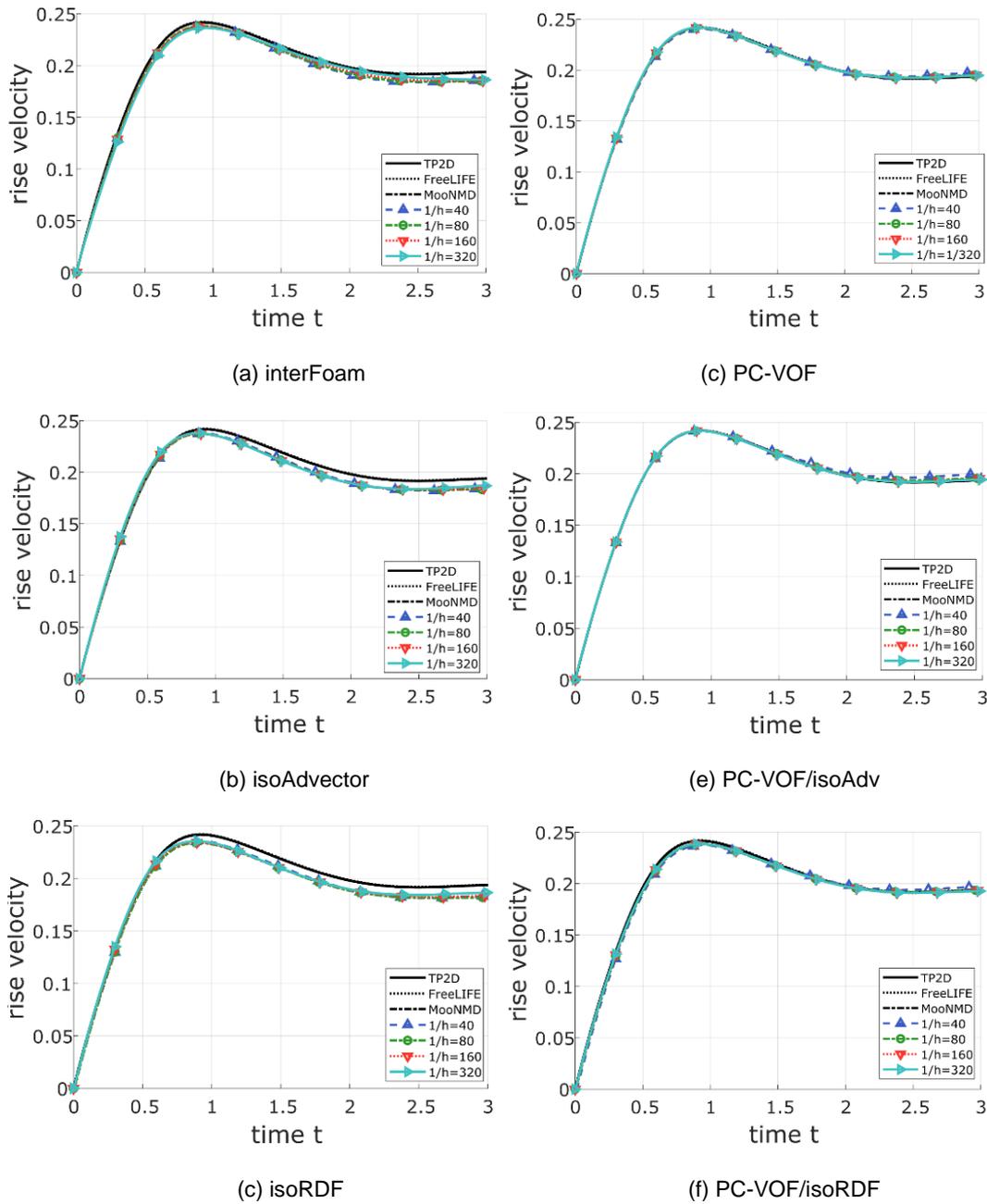


Figure 4.18 –Time variation of the mean rise velocity for the TC1 quadrangular mesh configuration.

For the mean rise velocity with triangular meshes (Fig. 4.19), a similar pattern to the other quantities in unstructured meshes is observed. The PC-VOF based methods had good agreement with benchmark data, particularly the coupled methods, that presented slightly improved results at later time steps than the original PC-VOF method. The isoAdvect, isoRDF and interFoam did not converge with mesh refinement.

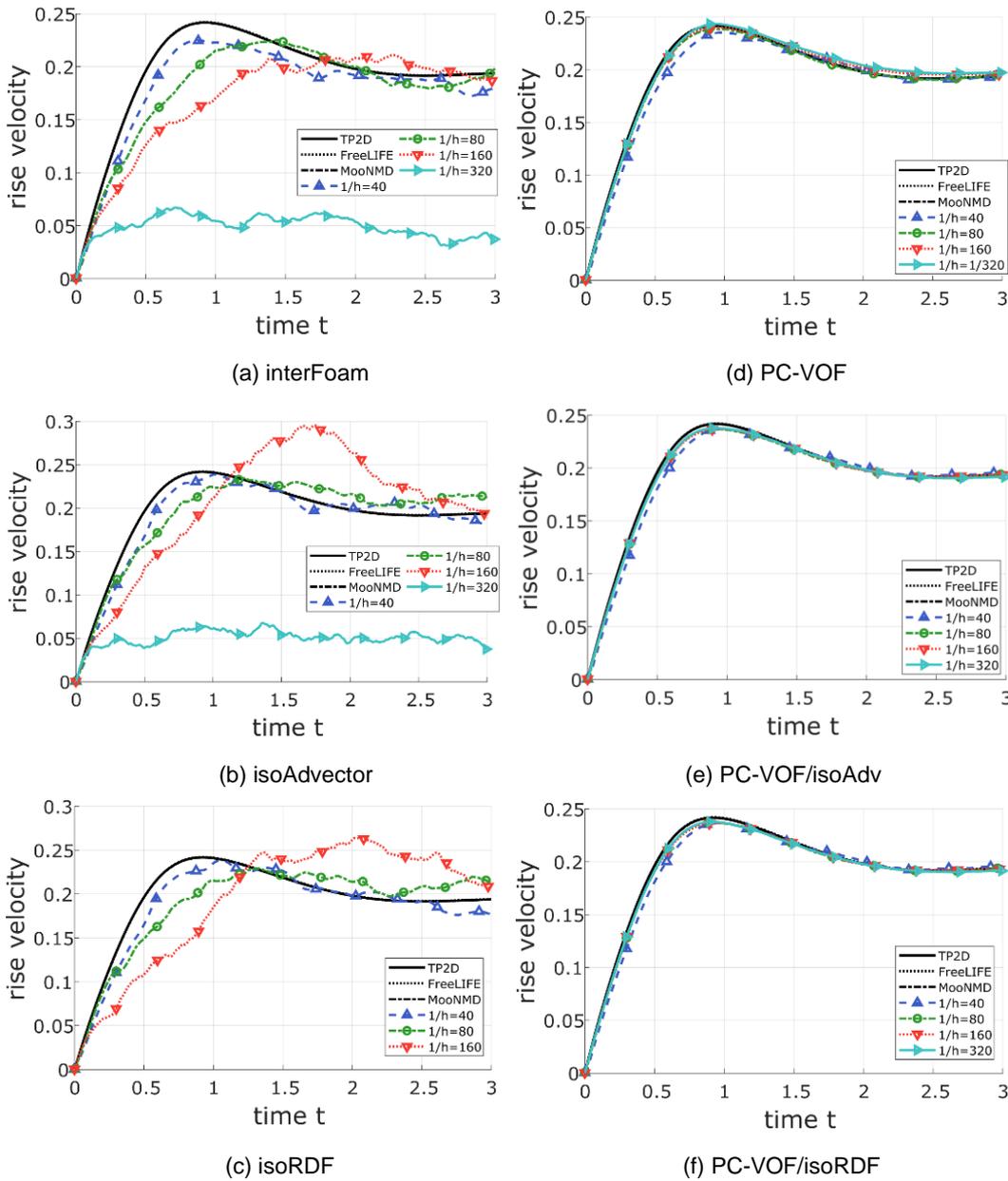
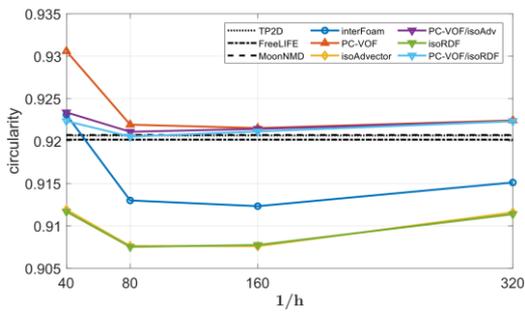


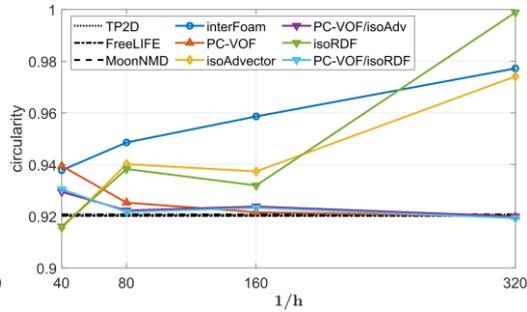
Figure 4.19 –Time variation of the mean rise velocity for the TC1 triangular mesh configuration.

Figure 4.20 shows a comparative grid test amongst all methods, and highlights the proximity between the solutions of the PC-VOF and the coupled methods. In particular, the circularity results for the coupled methods (Fig. 4.20-d and 4.20-e) were more accurate than the pure PC-VOF method (Fig. 4.20-d). The center of mass results for structured meshes show that PC-VOF (Fig. 4.20-d) had the most accurate predictions for coarser grids, eventually being surpassed by the PC-VOF/isoAdv coupled method. By analyzing the grid tests for triangular grids (Fig. 4.20-a, 4.20-b,

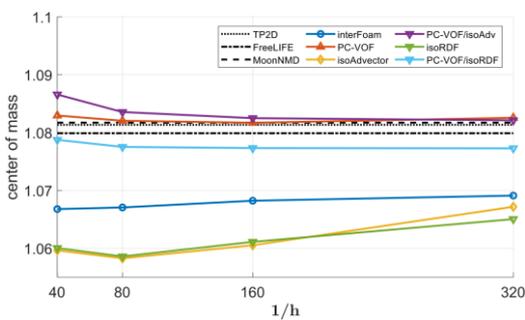
4.20-c), one can notice the lack of mesh convergence of the interFoam, isoAdvect and isoRDF cases.



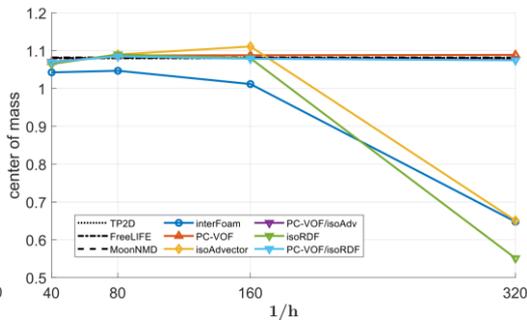
(a) Circularity quad. Mesh.



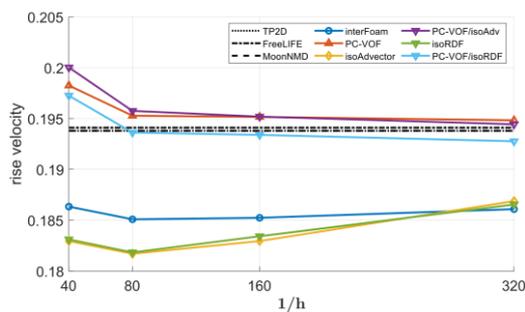
(b) Circularity, tri. Mesh.



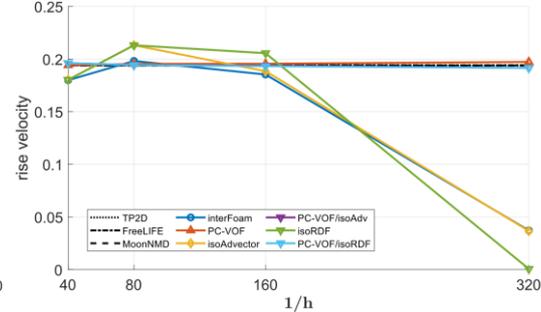
(c) Center of mass quad. Mesh.



(d) Center of mass, tri. Mesh.



(e) Rise velocity quad. Mesh.



(f) Rise velocity, tri. Mesh.

Figure 4.20 –Grid test comparison.

Table 4.5 shows the average errors in the prediction of the evaluated flow quantities against the three solutions provided in the benchmark data for the quadrangular and triangular meshes with $h = 1/320$ spacing. The errors for every benchmark solution provided by Hysing et al. (2009) were calculated, and an averaging process took place in order to provide a global error. Overall, for the quadrilateral mesh, the errors were very low for all methods. The coupled method PC-VOF/isoAdv presented the most accurate solution, particularly in center of mass prediction with an error of

0.09%. The results for the triangular mesh show an increase in the error magnitude for the interFoam, isoAdvector and isoRDF cases, and an increase in the disparity between those errors and the ones obtained by the PC-VOF base methods. The coupled methods achieved almost identical results which were superior to the original PC-VOF method.

Table 4.5 –Errors for benchmark quantities predictions with all methods for quadrangular and triangular grids of $h = 1/320$ spacing.

Methods	quadrangular grid			triangular grids		
	C	V_{rise}	\bar{Y}_C	C	V_{rise}	\bar{Y}_C
InterFoam	1.11%	4.02%	0.59%	6.16%	88.96%	39.81%
IsoAdvector	1.30%	3.61%	0.97%	5.82%	81.10%	39.81%
IsoRDF	1.48%	3.82%	0.99%	8.52%	81.10%	49.03%
PC-VOF	0.19%	0.46%	0.20%	0.13%	1.75%	0.74%
PC-VOF/isoAdv	0.09%	0.26%	0.20%	0.13%	1.24%	0.65%
PC -VOF/isoRDF	0.37%	0.57%	0.20%	0.13%	1.24%	0.65%

4.3.3 Test Case 2 – Rising Condition

Following the TC1 case, the rising bubble simulation with the TC2 configuration took place. The solutions for the rising bubble tests from the three solvers, TP2D, FreeLIFE and MooNMD were contemplated in this work for the TC2 as well.

Unlike the TC1 configuration, the present test case involves a less surface tension dominated flow, where inertial and gravitational forces play a more significant role. The bubble shape differs from the previous test case, since during ascension, the bubble takes a skirted shape and gains a more complicated topology. The possibility of bubble breakup is also high, given the elongation of the skirt area.

Given the diminished prominence of surface tension in the test case 2 configuration, the enhancements in curvature computation were not as influential in the overall results for the evaluated quantities. This effect is clearly seen in the bubble shape prediction shown in Fig. 4.21. The bubble shape of all methods were well in accordance with the benchmark data, both for quadrangular and triangular meshes.

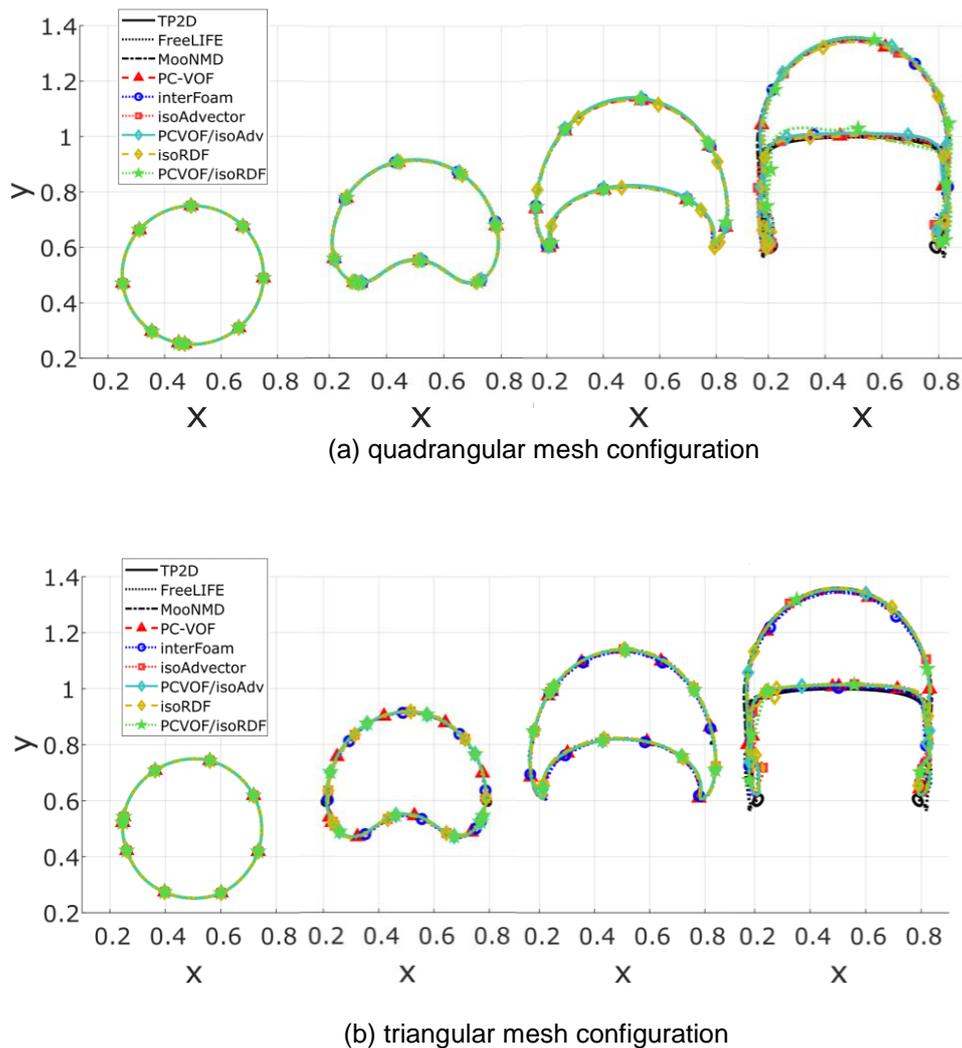


Figure 4.21 – Time evolution of the bubble shape for the TC2

The results for circularity are shown in figure 4.22, where the solution of the TP2D solver showed a deviation from the others, as it predicted bubble breakup in more advanced time steps of the simulation. The interFoam (Fig. 4.22-a) and PC-VOF (Fig. 4.22-d) had similar results and were more aligned with the FreeLIFE and MoonMD solutions.

IsoAdvectord (Fig. 4.22-b) and isoRDF (Fig. 4.22-c) also presented good agreement to the benchmark data, however, for the $h = 1/160$ grid, they predicted breakup, but at a different time-step from the TP2D solution. The PC-VOF coupled methods (Fig. 4.22-e, 4.22-f) had similar results to the original PC-VOF, and had a more stable circularity variation at $t > 2.75s$, where interFoam and PC-VOF showed oscillations.

The triangular mesh results (Fig. 4.23) were, again, very similar to the structured mesh. The main difference that is worth noting is the prediction of breakup by the PC-VOF based methods (Fig. 4.23-d, 4-23-e, 4.23-f) with the $h = 1/40$ grid that did not occur on structured meshes. The results as a whole were close to benchmark data.

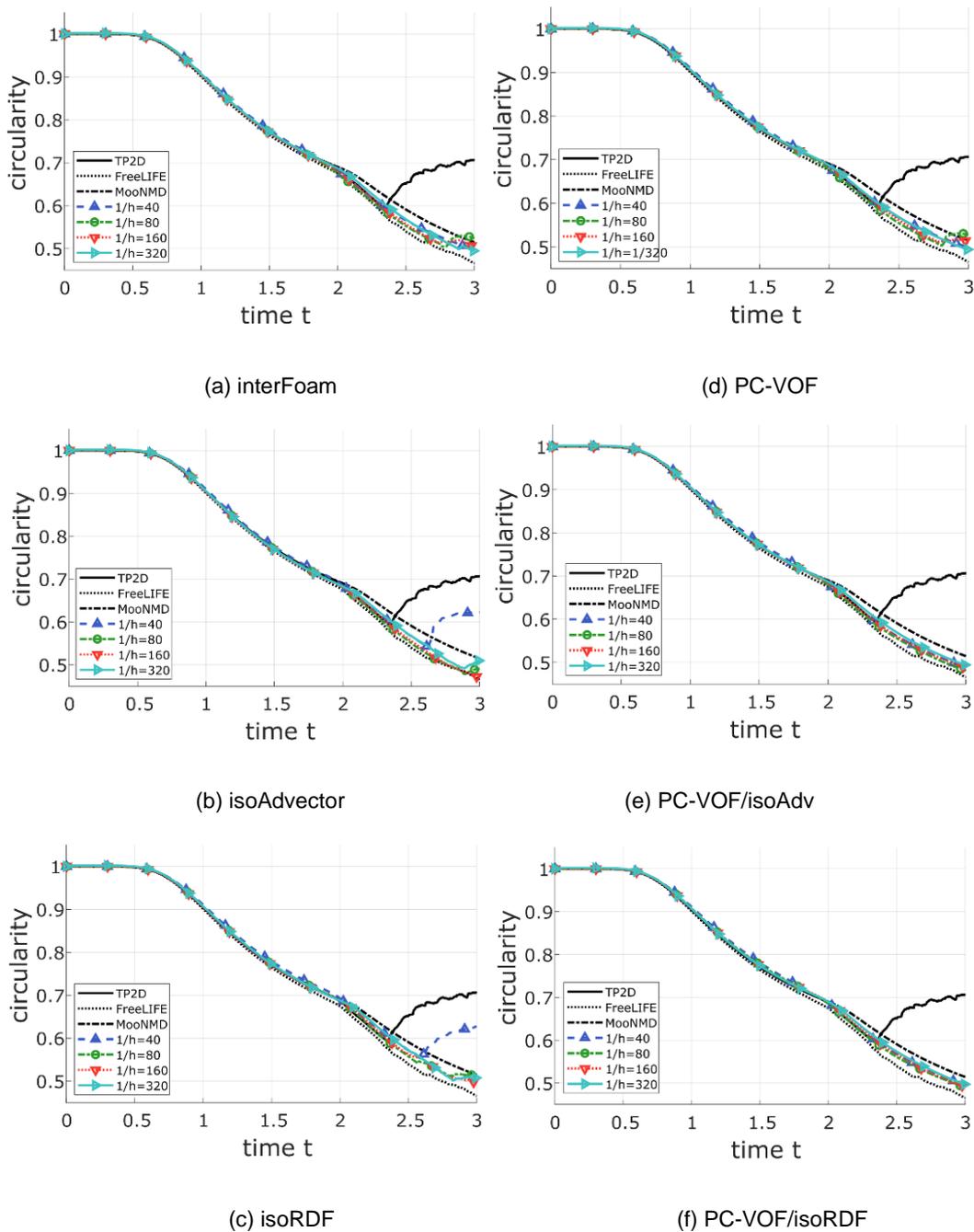


Figure 4.22 –Time variation of curvature for the TC2 quadrangular mesh configuration.

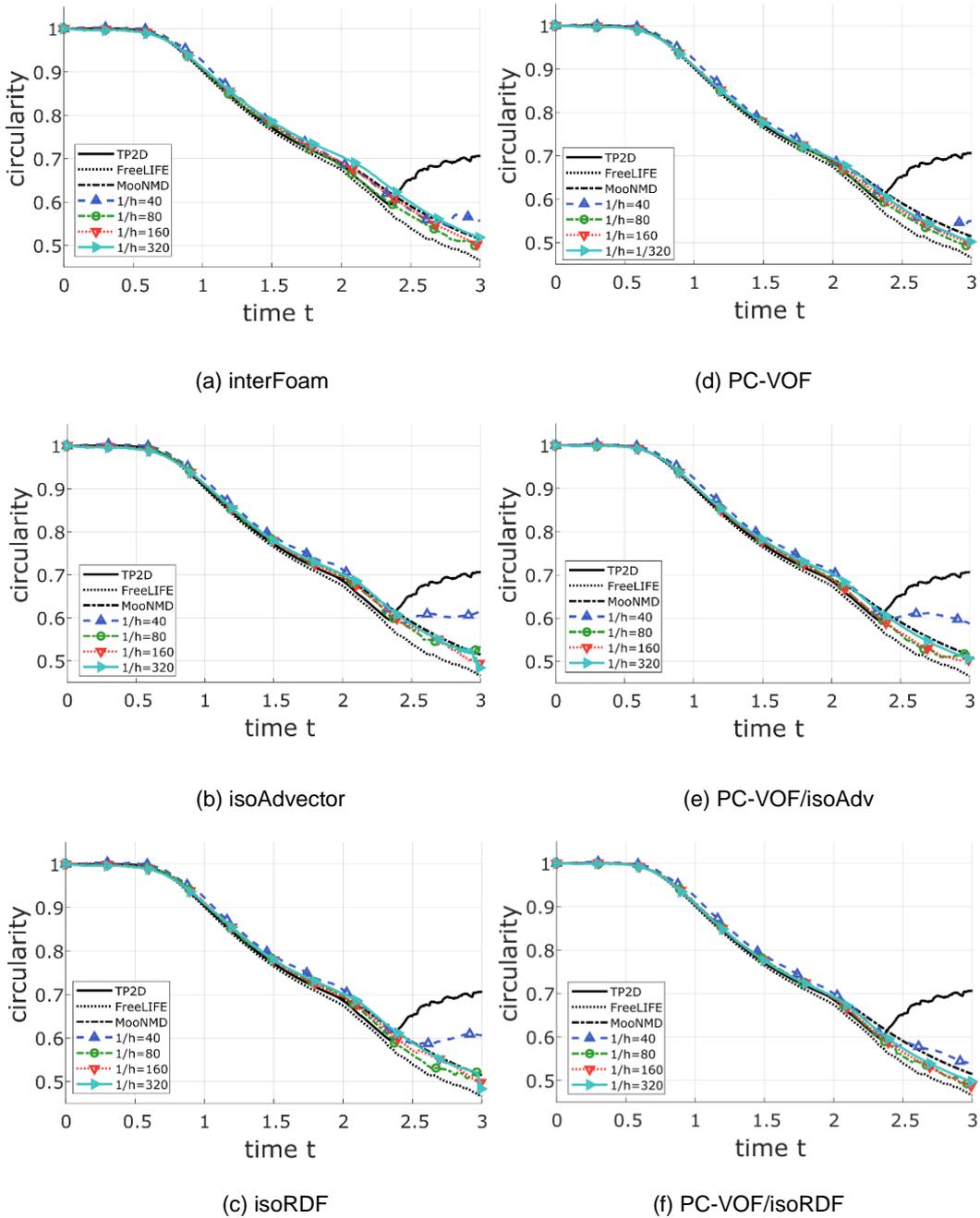


Figure 4.23 –Time variation of curvature for the TC2 triangular mesh configuration.

The center of mass results (Fig. 4.24) follow the pattern of the other quantities. All methods present similar solutions, where a small deviation from benchmark data occur at $t > 1.5s$. The PC-VOF based methods (Fig. 4.24d, 4.24-e, 4.24-f) did not improve the solutions. Results for the triangular mesh present the same behavior, and are displayed in Fig. 4.25.

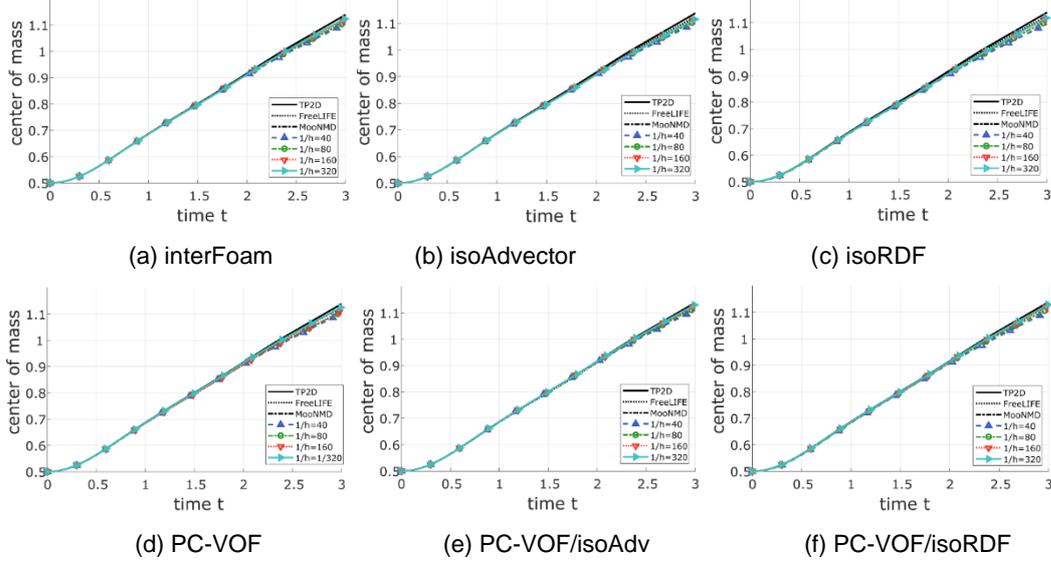


Figure 4.24 – Time variation of the center of mass for the TC2 quadrangular mesh configuration.

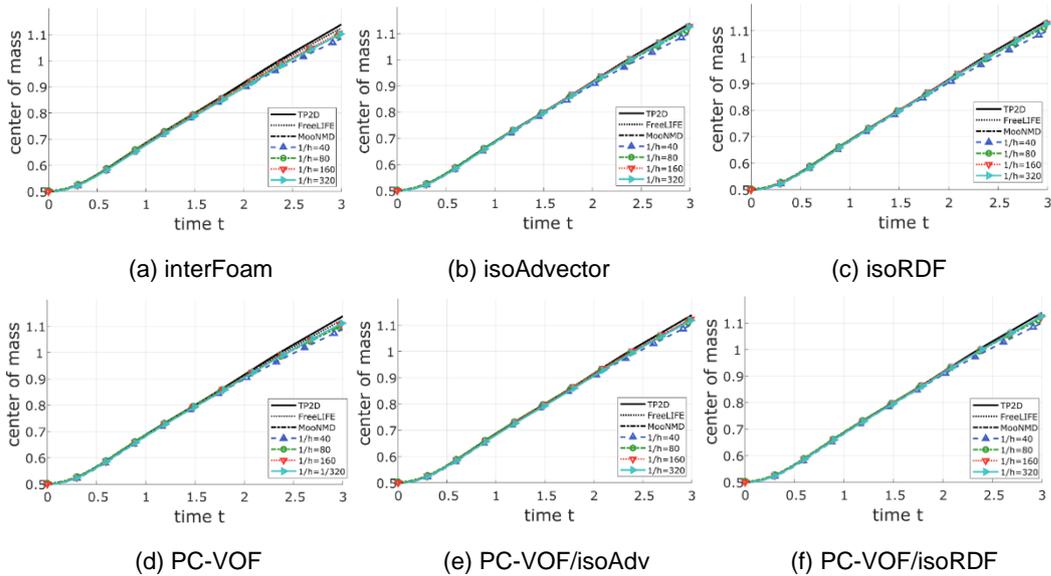


Figure 4.25 – Time variation of the center of mass for the TC2 triangular mesh configuration.

Regarding the rise velocity for quadrangular grids (Fig. 4.26), an almost identical pattern was observed for all methods. A deviation from benchmark data was detected once the slope of the V_{rise} curve decreased, and a smooth oscillating pattern takes place. Although finer meshes were able to more accurately predict those oscillations, the results were unable to adhere to the benchmark curve, unlike for other quantities. The isoRDF based methods (Fig. 4.26-e, 4.26-f) showed an oscillatory behavior at later time steps ($t > 2$), particularly for the $h = 1/160$ grid in the coupled method.

For the triangular mesh cases (Fig. 4.27), a similar pattern was observed, although the PC-VOF based methods had a smoother solution.

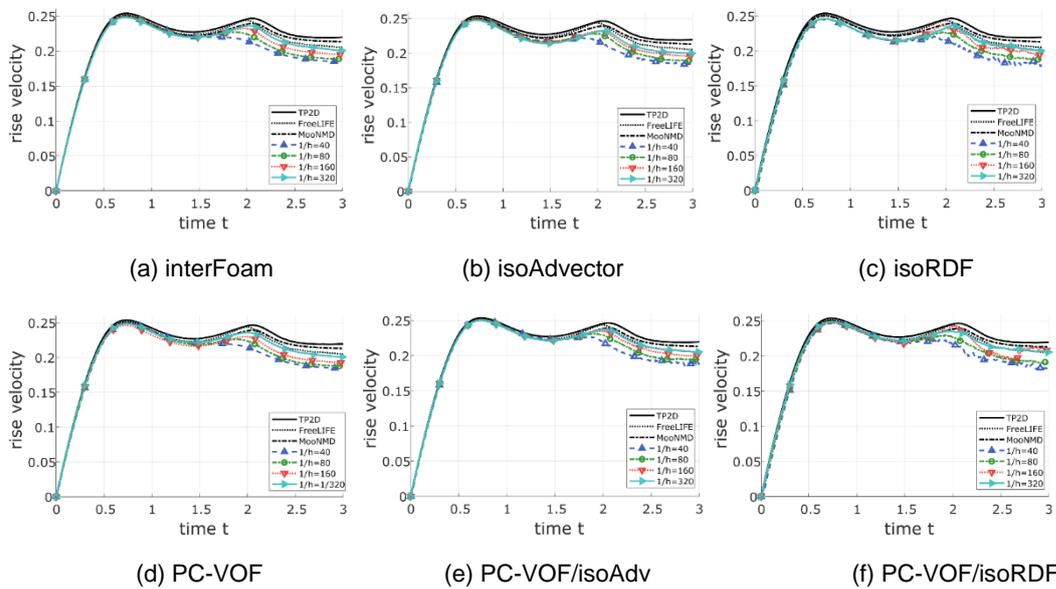


Figure 4. 26 - Time variation of the mean rise velocity for the TC2 quadrangular mesh configuration.

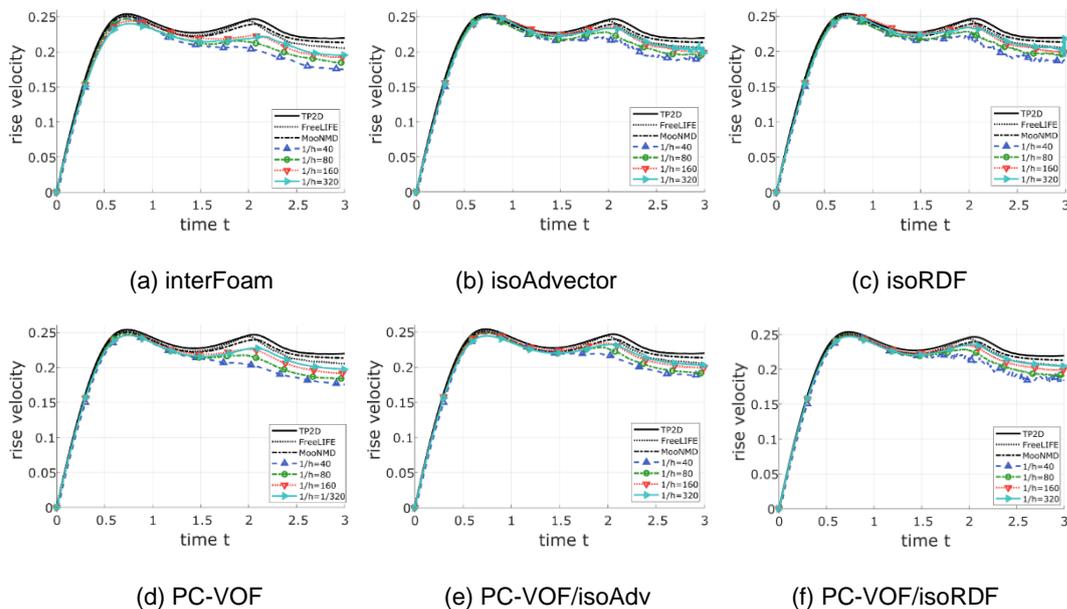


Figure 4.27 –Time variation of the mean rise velocity for the TC2 triangular mesh configuration.

Figure 4.28 displays a comparative grid test for all methods contemplated in the present work. The results shown further validate the assertion that the surface tension modelling imposes a minor influence in the accuracy of the obtained solutions. In fact, the Test Case 2 simulations appear to be more responsive to improvements in the advection methods.

For circularity C in quadrangular meshes (Fig 4.28-a) all methods converged to similar results, and had good agreement to the MoonMD and FreeLIFE solutions, whereas for the center of mass (Fig 4.28-c) and rise velocity (Fig 4.28-e) the results were more closely aligned to the FreeLIFE solution. For triangular meshes (Fig. 4.28-b, 4.28-d, 4.28-f) the methods based on the isoAdvector and isoRDF schemes held the most accurate solutions for center of mass and rise velocity. Both interFoam and PC-VOF employ the standard MULES advection algorithm, which is known to be less accurate than other algebraic VOF schemes as well as geometric advection schemes. The coupled methods also obtained accurate results, which were on par with the ones presented by the original isoAdvector and isoRDF.

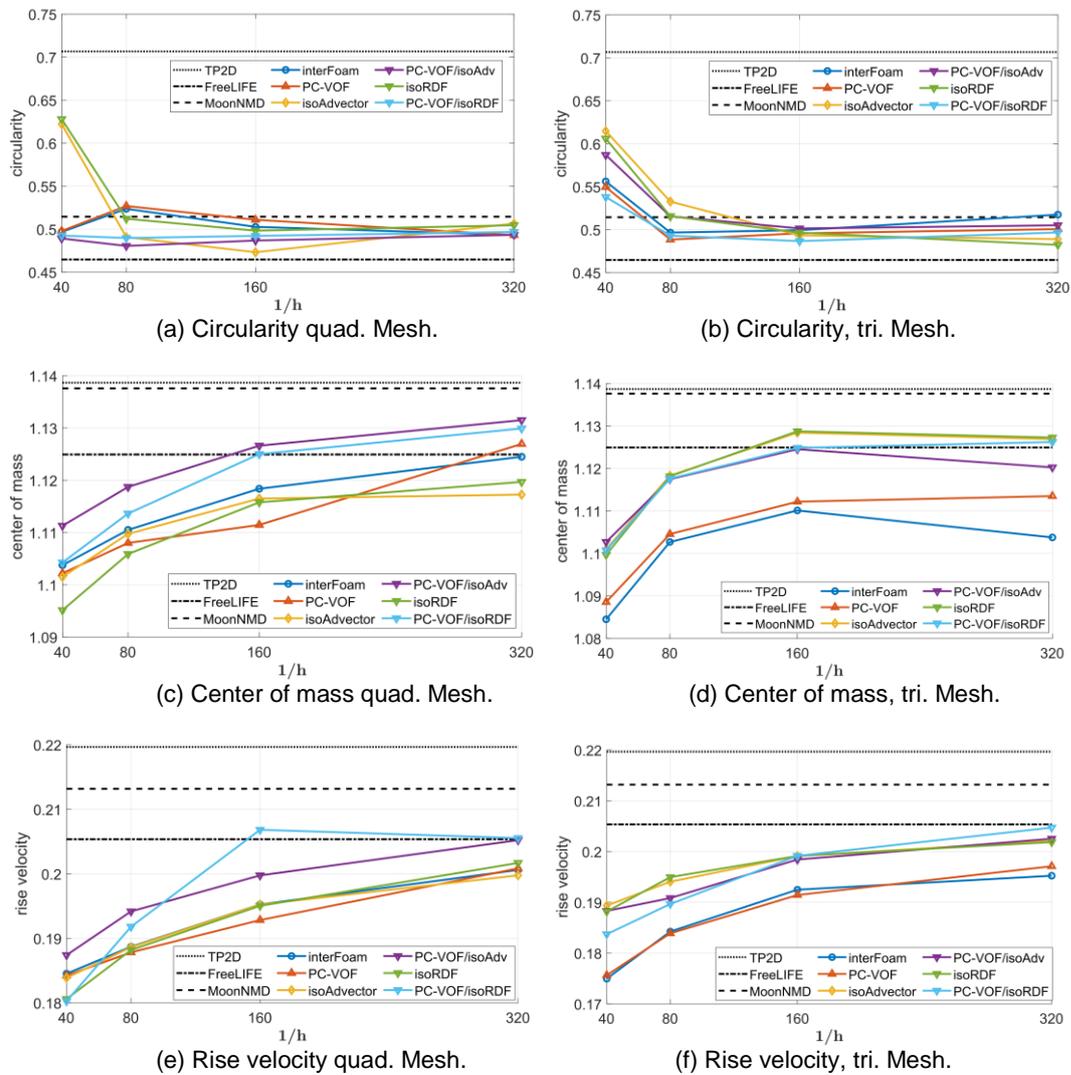


Figure 4.28 –Grid test comparison.

Table 4.6 displays the errors for the three flow quantities considered and all tested methods. Since the solution by TP2D significantly deviated from the other codes in circularity C prediction, it was not considered in the error calculation for C . The same procedure applied to the error estimation of TC1 was done for the TC2 configuration.

In quadrangular meshes, the most accurate results were once again achieved by the couple methods, PC-VOF/isoAdv and PC-VOF/isoRDF. However, the disparity was not as high as in the TC1 cases. For triangular grids, although the original isoAdvector and isoRDF obtained the most precise values for circularity and center of mass \vec{Y}_c , their predictions of rise velocity were inferior to the PC-VOF/isoRDF solution by a greater margin.

Table 4.6 –Errors for benchmark quantities predictions with all methods for quadrangular and triangular grids of $h = 1/320$ spacing.

Methods	quadrangular grid			triangular grids		
	C_{min}	V_{rise}	\vec{Y}_c	C_{min}	V_{rise}	\vec{Y}_c
InterFoam	5.16%	5.65%	0.88%	5.96%	8.19%	2.64%
IsoAdvector	5.27%	6.02%	1.5%	5.08%	4.94%	0.73%
IsoRDF	5.24%	5.13%	1.23%	5.01%	5.04%	0.73%
PC-VOF	5.13%	5.55%	0.73%	5.2%	7.29%	1.85%
PC-VOF/isoAdv	5.13%	3.48%	0.62%	5.25%	4.75%	1.23%
PC -VOF/isoRDF	5.16%	3.37%	0.65%	5.16%	3.72%	0.76%

5. Conclusion

The present work aimed to evaluate the performance of different methods within the VOF framework against test cases with either analytical solutions or benchmark data for comparison. The preliminary test case results were underwhelming in the sense that the square interface cases showed virtually no difference between the tested methods, and the oscillating drop showed poor results for the coupled methods. In fact, the oscillating drop case requires further investigation as to the reason behind the under performance of the coupled methods in relation to the original PC-VOF.

The benchmark test case results were very promising. TC1 highlighted the importance of the curvature computation in surface tension modelling, as the results of PC-VOF based methods were far greater than the ones obtained by `interFoam`, `isoAdvector` and `isoRDF`, particularly for the triangular meshes. On the other hand, TC2 proved the importance of an enhanced advection algorithm, as the results presented by the `isoAdvector` and `isoRDF` based methods were the most accurate ones.

The coupled methods excelled in almost every comparison carried out in this work, and can be recommended for a wide variety of case configurations. The relaxing bubble simulation exposed a major issue with the `interFoam`, `isoAdvector` and `isoRDF` related to their pressure fields. Coupling these methods with PC-VOF mitigates this issue almost entirely, and, therefore, constitute a feasible and more stable alternative to their original constituent methods.

Further investigation into their performance for three-dimensional simulations remains to be pursued in order to assert their superior capabilities. Many modifications can be performed in order to ensure a more optimized solver with a tighter coupling between the methods.

Bibliography

- Albadawi, A., Donoghue, D. B., Robinson, A. J., Murray, D. B., & Delauré, Y. (2013). Influence of surface tension implementation in Volume of Fluid and coupled Volume of Fluid with Level Set methods for bubble growth and detachment. *International Journal of Multiphase Flow*, 53, 11-28.
- Aulisa, E., Manservigi, S., Scardovelli, R., & Zaleski, S. (2007). Interface reconstruction with least-squares fit and split advection in three-dimensional cartesian geometry. *Journal of Computational Physics*, 225, 2301-2319.
- Boris, J. P., & Book, D. L. (1973). Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works. *Journal of Computational Physics*, 11, pp. 38-69.
- Brackbill, J., Kothe, D., & Zemach, C. (1992). A continuum method for modeling surface tension. *Journal of Computational Physics*, 100, 335-354.
- Brennen, C. (2003). *Fundamentals of Mutiphase Flow*. Pasadena, CA: Oxford University Press, Inc.
- Cano-Lozano, J. C., Bolaños-Jiménez, R., Gutiérrez-Montes, C., & Martínez-Bazán, C. (2014). The use of Volume of Fluid technique to analyze multiphase flows: Specific case of bubble rising in still liquids. *Applied Mathematical Modelling*, 39, 3290-3305.
- Cummins, S. J., Francois, M. M., & Kothe, D. B. (2005). Estimating the curvature from volume fractions. *Computers & Structures*, 425-434.
- Damián, S. M. (2013). *An Extended Mixture Model for the Simultaneous Treatment of Short and Long Scale Interfaces*. Doctorate Thesis, Universidad Nacional del Litoral.
- de Melo, J. R. (1995). *Simulação numérica de escoamentos bifásicos com interface*. Msc. Thesis, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brazil.
- Debar, R. (1974). *Fundamentals of the KRAKEN code*. LLNL Tech. Rep. #UCIR-760.
- Evrard, F., Denner, F., & Wachem, v. (2017). Estimation of curvature from volume fractions using parabolic reconstruction on two-dimensional unstructured meshes. *Journal of Computational Physics*, 351, 271-294.

- Francois, M. M., Cummins, S. J., Dendy, E. D., Kothe, D. B., & Williams, M. W. (2006). A balanced force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *Journal of Computation Physics*, 213(1), 141-173.
- Fyfe, D. E., Oran, E. S., & Fritts, M. (1988). Surface Tension and Viscosity with Lagrangian hydrodynamics on triangular mesh. *Journal of Computational Physics*, 76(2), 349-384.
- Gopala, V., & van Wachem, B. (2008). Volume of fluid methods for immiscible-fluid and free-surface flows. *Chemical Engineering Journal*, 141(1-3), 204-221.
- Hardt, S., & Wondra, F. (2008). Evaporation model for interfacial flows based on a continuum-field representation of the source terms. *Journal of Computational Physics*, 227, 5871-5895.
- Harlow, F. H., & Welch, J. E. (1965). Numerical Calculation of Time Dependent Viscous Incompressible Flow of Fluid with Free Surface. *Physics of Fluids*, 8(12), 2182-2189.
- Hirt, C. W., & Nichols, B. D. (1981). Volume of Fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39, 201-225.
- Hysing, S., S, Turek, Kuzmin, D., Parolini, N., Burman, E., . . . Tobiska, L. (2009). Quantitative benchmark computations of two-dimensional bubble dynamics. *International Journal for numerical methods in fluids*, 60, 1259-1288.
- Ivey, C. B., & Moin, P. (2017). Conservative and bounded volume-of-fluid advection on unstructured grids. *Journal of Computational Physics*, 350, 387-419.
- Kassar, B. d. (2016). *Numerical simulation of multiphase flows with enhanced curvature computation by point-cloud sampling*. Doctorate Thesis, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.
- Kassar, B. d., Carneiro, J. N., & Nieckele, A. O. (2018). Curvature computation in volume-of-fluid method based on point-cloud sampling. *Computer Physics Communications*, 222, 189-208.
- Klostermann, J., Schaake, K., & Schwarze, R. (2012). Numerical simulations of a single rising bubble by VOF with surface compression. *International Journal for Numerical Methods in Fluids*, 71, 960-982.
- Liu, L., Yan, H., & Zhao, G. (2015). Experimental studies on the shape and motion of air bubbles in viscous liquids. *Experimental Thermal and Fluid Science*, 62, 109-121.

- Lopez, J., Gómez, P., Hernández, J., & Faura, F. (2005). An improved PLIC-VOF method for tracking thin fluid structures in incompressible two-phase flows. *Journal of Computational Physics*, 208(1), 51-74.
- Mencinger, J., & Zun, I. (2011). A PLIC-VOF method suited for adaptive moving grids. *Journal of Computational Physics*, 230(3), 644-663.
- Mirjalili, S., Jain, S. S., & Dodd, M. S. (2017). Interface-capturing methods for two-phase flows: An overview and recent developments. *Center for Turbulence Research: Annual Research Briefs*, 117-135.
- Noh, W. F., & Woodward, P. (1976). SLIC (simple lin interface calculation). *Proceedings, Fifth International Conference on Fluid Dynamics, Lecture Notes in Physics*, 59, 330-340.
- OpenFOAM. (2018). *The Open Source CFD Toolbox - User Guide. 6 edn.* Retrieved from The OpenFOAM Foundation.
- Osher, S., & Sethian, J. A. (1988). Fronts propagating with a curvature dependent speed:: algorithms based on Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79(1), 12-49.
- Patel, H. V., Kuipers, J. A., & Peters, E. A. (2017). Computing interface curvature from volume fractions: A hybrid approach. *Computers and Fluids*, 161, 74-88.
- Pilliod, J. E., & Puckett, E. G. (2004). Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics*, 199, 465-502.
- Popinet, S. (2003). Gerris: A tree-based adaptive solver for the incompressible Euler equations in complex geometries. *Journal of Computational Physics*, 190, 572-600.
- Popinet, S. (2009). An accurate adaptive solver for surface-tension-driven interfacial flows. *Journal of Computational Physics*, 228, 5838-5866.
- Popinet, S. (2018). Numerical models of surface tension. *Annu. Rev. Fluid Mech.*, 50, 49-75.
- Prosperetti, A., & Tryggvason, G. (2007). *Computational Methods for Multiphase Flow*. Cambridge University Press.
- Rayleigh, L. (1879). On the Capillary Phenomena of Jets. *Proceedings of the Royal Society of London*, 29, 71-97.
- Renardy, Y., & Renardy, M. (2002). PROST: A Parabolic Reconstruction of Surface Tension for the Volume-of-Fluid Method. *Journal of Computational Physics*, 183, 400-421.
- Rider, W. J., & Kothe, D. B. (1998). Reconstructing volume tracking. *Journal of Computational Physics*, 141, 112-152.

- Roenby, J., Bredmose, H., & Jasak, H. (2016). A computational method for sharp interface advection. *Royal Society Open Science*.
- Saincher, S., & Banerjee, J. (2014, Dezembro 10). A Redistribution-Based Volume-Preserving PLIC-VOF Technique. *Numerical Heat Transfer, Part B: Fundamentals*, 67, 338-362.
- Scheufler, H., & Roenby, J. (2019). Accurate and efficient surface reconstruction from volume fraction data on general meshes. *Journal of Computational Physics*.
- Skarysz, M., Garmory, A., & Dianat, M. (2018). An iterative interface reconstruction method for PLIC in general convex grids as part of a Coupled Level Set Volume of Fluid solver. *Journal of Computational Physics*, 368, 254-276.
- Su, J., Nascimento, F. R., Azevedo, M. B., Faccini, J. L., & Moreira, M. L. (2018). COMPUTATIONAL SIMULATION OF TAYLOR BUBBLES MOTION IN A STAGNANT LIQUID INSIDE A VERTICAL COLUMN. *17th Brazilian Congress of Thermal Sciences and Engineering*. Águas de Lindóia, SP, Brazil.
- Sussman, M. (2003). A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics*, 187, 110-136.
- Sussman, M., & Puckett, E. G. (2000). A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows. *Journal of Computational Physics*, 162, 301-337.
- Tryggvason, G., Scardovelli, R., & Zaleski, S. (2011). *Direct numerical simulations of gas-liquid multiphase flow*. Cambridge University Press.
- Ubbink, O. (1997). *Numerical prediction of two fluid systems with sharp interfaces*. Tese de Doutorado, Imperial College, Londres.
- Univerdi, S. O.; Tryggvason, G. (1992). A Front Tracking Method fo Viscous Incompressible Flows. *Journal of Computational Physics*, 100, 25-37.
- Yang, X., James, A. J., Lowengrub, J., Zheng, X., & Cristini, V. (2006). And adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids. *Journal of Computational Physics*, 217, 364-394.
- Youngs, D. L. (1982). Time-dependent multi-material flow with large fluid distortion. *Numer. Methods Fluid Dyn.*, 1, 41-51.
- Zalesak, S. (1979). Fully multidimensional Flux-Corrected Transport algorithms for fluids. *Journal of Computational Physics*, 31, 335-362.