

AMERICAN UNIVERSITY OF BEIRUT

ON THE IMPLEMENTATION OF SURFACE  
TENSION WITHIN A VOF FRAMEWORK

by  
ADAM NAZIH FARES

A thesis  
submitted in partial fulfillment of the requirements  
for the degree of Master of Engineering  
to the Department of Mechanical Engineering  
of the Maroun Semaan Faculty of Engineering and Architecture  
at the American University of Beirut


Beirut, Lebanon  
August 2023

AMERICAN UNIVERSITY OF BEIRUT


ON THE IMPLEMENTATION OF SURFACE  
TENSION WITHIN A VOF FRAMEWORK

by  
ADAM NAZIH FARES

Approved by:

  
\_\_\_\_\_  
Dr. Fadel Moukalled, Professor  
Department of Mechanical Engineering

Advisor

  
\_\_\_\_\_  
Dr. Marwan Darwish, Professor  
Department of Mechanical Engineering

Member of Committee

  
\_\_\_\_\_  
Dr. Kamel Abou Ghali, Professor  
Department of Mechanical Engineering

Member of Committee

Date of thesis presentation: August 2<sup>nd</sup>, 2023



## ACKNOWLEDGEMENTS

I express my sincere gratitude to my advisor, Prof. Moukalled, for his guidance, support, and encouragement throughout the research process. His expertise and constructive feedback shaped this thesis and enhanced my academic growth.

I am deeply thankful to my family and friends for their constant love, understanding, and encouragement, which motivated me throughout my academic journey. I extend my heartfelt appreciation to the lab members for their collaboration, encouraging discussions, and the vibrant research environment they fostered. Their collective expertise and valuable input greatly contributed to the development and improvement of this thesis.

All these contributions played an integral role in the completion of this master thesis, and for that, I am truly grateful.

# ABSTRACT OF THE THESIS OF

Adam Nazih Fares

for

Master of Engineering

Major: Mechanical Engineering

Title: On the Implementation of Surface Tension Within a VOF Framework

A balanced-force algorithm for predicting free surface flow with surface tension on general curvilinear coordinate systems is presented in this thesis. The well-established Continuum Surface Force (CSF) method is employed to model surface tension forces. One of the primary challenges of the CSF method is the generation of spurious currents in the vicinity of the interface. To address this issue, the presented algorithm reduces imbalances between pressure gradients and surface tension forces to the solver's tolerance, effectively eliminating the first source of spurious currents.

The remaining task is to adequately estimate curvature, which cannot be directly obtained using the abruptly varying volume fraction field. Three well known curvature estimation techniques within the Volume of Fluid (VOF) framework are implemented: Laplacian Filter, Convolution Method, and Height Functions. The algorithm is then validated through several benchmark cases, and the parameters associated with each of the curvature estimation techniques are refined to enhance the solution's accuracy.

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS .....	1
ILLUSTRATIONS .....	6
TABLES .....	8
NOMENCLATURE .....	9
INTRODUCTION .....	11
LITERATURE REVIEW .....	15
GOVERNING EQUATIONS .....	19
3.1. General Solution on Collocated Grids .....	19
3.2. Volume Fraction Equation.....	21
3.3. Balanced Force Algorithm.....	22
3.3.1 Face-Centered Formulation .....	22
3.3.2. Redistributed Cell-Centered Formulation.....	23
3.3.3. Present Formulation.....	24
SURFACE TENSION AND CURVATURE MODEL .....	27
4.1. CSF Model.....	27
4.2. Curvature Equations .....	28
4.3. Laplacian Filter .....	29
4.4. Convolution .....	30

4.4.1 Choice of Kernel .....	31
4.4.2. Normalization Factor .....	32
4.4.3. Support Length .....	33
4.5. Height Function .....	34
4.5.1. Identify the Interfacial Area.....	34
4.5.2. Extend Computational Stencil .....	34
4.5.3. Correct VOF in Extended Stencil .....	35
4.5.4. Choose Height Function Direction .....	36
4.5.5. Calculate Height Function and Curvature .....	37
4.5.6. Distribute Curvature .....	37
4.5.7. Bounding Height Function.....	38
<b>NUMERICAL TEST CASES .....</b>	<b>39</b>
5.1. Inviscid Stationary Fluid Column.....	39
5.2. Inviscid Static Drop at Equilibrium.....	41
5.2.1. Exact Curvature .....	42
5.2.2. Curvature Errors .....	47
5.2.3. Laplacian Filter .....	48
5.2.4. Convolution .....	49
5.2.5. Height Functions.....	51
5.2.6. Comparing Different Techniques .....	53
5.2.7. Time Evolution .....	54
5.3. Static Bubble at Equilibrium.....	54
5.4. Breaking Dam of Water .....	55
5.5. Forced Sloshing in Water Tank .....	61

CONCLUSION .....	65
BIBLIOGRAPHY .....	66



# ILLUSTRATIONS

## Figure

1.1: Moving mesh (left) versus fixed mesh (right) [15] .....	12
4.1: Neighboring cells used in Laplacian Filter .....	30
4.2: Kernel weight distribution as a function of distance from its center .....	33
5.1: Stationary fluid column volume fraction field.....	40
5.2: Stationary fluid column maximum velocity as a function of gridratio.....	40
5.3: Computed pressure with different CSF models for the inviscid static drop case when the exact curvature is specified. The grid is uniform and of resolution $R/h = 10$ . The density ratio is 103. Redistributed left and cell centered right. ...	44
5.4: Errors in velocity magnitude as a function of grid spacing. ....	45
5.5: Errors in calculated pressure as a function of density ratio. ....	45
5.6: Errors in velocity magnitude as a function of grid spacing. ....	46
5.7: Errors in calculated pressure as a function of grid spacing. ....	46
5.8: Error in calculated pressure as a function of grid spacing. ....	46
5.9: Error in maximum velocity as a function of grid spacing. ....	46
5.10: Maximum residuals obtained with redistributed CSF (left) and cell centered CSF (right) models. ....	47
5.11: Velocity magnitude as a function of grid spacing when curvature is numerically estimated directly from VOF field.....	48
5.12: Errors in radius of curvature estimation as a function of grid spacing.....	48
5.13: Errors in pressure jump as a function of grid spacing. ....	48
5.14: Effect of Laplacian smoothing on curvature estimation.....	49
5.15: Effect of Laplacian smoothing on spurious current generation.....	49
5.16: Errors in Curvature estimation as a function of grid spacing using Laplacian Filter. ....	49
5.17: Spurious current generation as a function of grid spacing using Laplacian Filter.....	49
5.18: Errors in curvature estimation as a function of support length.....	51

5.19:Errors in curvature estimation as a function of grid spacing. ....	51
5.20: Spurious current generation as a function of grid spacing. ....	51
5.21:Errors in total pressure jump as a function of grid spacing. ....	51
5.22: Errors in curvature as a function of grid spacing using HF. ....	52
5.23: Errors in curvature estimation as a function of stencil size using HF. ....	52
5.24: Errors in total pressure jump as a function of grid spacing using HF. ....	52
5.25: Spurious current generation as a function of grid spacing using HF. ....	52
5.26: Curvature errors as a function of grid spacing. ....	53
5.27: Spurious current generation as a function of grid spacing. ....	53
5.28: Errors in total pressure jump as a function of grid spacing. ....	53
5.29: Maximum velocity magnitude as a function of time. ....	55
5.30: Total kinetic energy as a function of time. ....	55
5.31: Time evolution of volume fraction field for both cases excluding surface tension (left column- dark blue) and including surface tension (right column- lighter blue). ....	60
5.32: Non-dimensional distance to leading edge versus non-dimensional time. ....	61
5.33: Non-dimensional residual column height versus non-dimensional time... ....	61
5.34: Schematic of water sloshing problem. ....	62
5.35: Water height at the left-hand-side of the tank ....	62
5.36: Time evolution of volume fraction field for both cases excluding surface tension (left column- dark blue) and including surface tension (right column- lighter blue). ....	64

## TABLES

### Table

1. Errors in calculated pressure as a function of grid spacing after one and hundredth timestep.....	41
2. The maximum velocity magnitude and pressure jump after one time step of $10^{-6}$ .....	43
3. Spurious current generation at $t = 10 - 3s$ .....	54

## NOMENCLATURE

### Abbreviations

Computational Fluid Dynamics	CFD
Finite Volume Method	FVM
Volume of Fluids	VOF
Continuum Surface Force	CSF
Continuum Surface Stress	CSS
Sharp Surface Force	SSF
Height Function	HF
Total kinetic Energy	TKE

### Fields and Parameters

Velocity	$\mathbf{V}$
Velocity- X component	$U$
Velocity- Y component	$V$
Pressure	$P$
Mass flow rate	$\dot{m}$
Volume fraction	$\alpha$
Smoothed volume fraction	$\tilde{\alpha}$
Density	$\rho$
Viscosity	$\mu$
Surface tension coefficient	$\sigma$
Gravitational acceleration	$g$
2D Rectangular coordinates	$X ; Y$
Time	$t$
Position vector	$\mathbf{r}$
Body force	$\mathbf{B}$
Interface normal vector	$\mathbf{n}$
Interface unit normal	$\hat{\mathbf{n}}$
Face normal vector	$\mathbf{S}$
Interface curvature	$\kappa$
Element volume	$V$
Number of phases	$n$
Number of elements	$N$
Drop radius	$R$
Convolution Kernel	$K$
Grid spacing	$h$
Horizontal grid spacing	$\Delta x$
Vertical grid spacing	$\Delta y$
Pressure jump across interface	$\Delta P$
Exact pressure	$P_e$
Convolution support length	$d$
Geometric interpolation factor	$g_F$
Constant= 1e-6	$\epsilon_1$
Constant= 1e-3	$\epsilon_2$

**Superscripts**

Phase index	k
Previous time step value	o
Momentum	v
Pressure correction	p'

**Subscripts**

Element	C
Neighbor element	F
Face	f
Boundary face	b

# CHAPTER I

## INTRODUCTION

With the ever-growing applications in a wide range of technologies, fluid flow analysis has become increasingly essential. However, the complex equations governing fluid flow have no analytical solution, thus numerical tools, such as Computational Fluid Dynamics (CFD), are used to predict the flow fields [1]. At the heart of these equations is the Navier-Stokes equation that accurately models flow phenomena in diverse regimes [2]. The Finite Volume Method (FVM) is one of the well-established conservative numerical methods used to implement CFD, in which the integral form of the nonlinear partial differential equations is discretized into a linear system of algebraic equations to be solved numerically [2, 3].

When multiple phases exist in the domain, each of the phases requires its own set of conservation laws, however, the physics is unchanged, and the same equations hold [4]. If the normal and tangential stresses at the interface between the phases are negligible, then the same set of conservation equations is shared between all phases [5], and this class of multiphase flow is referred to as free-surface flow. The widespread of such flow phenomena in engineering applications has pushed the research community to develop accurate and robust free-surface flow solvers [6, 7, 8, 9]. That said, numerical techniques for the solution of free surface flow can be classified under two general categories: moving mesh (Lagrangian method) and fixed mesh (Eulerian method) [10, 11]. In a moving mesh approach (figure 1.1- left), distinct boundary-fitted grids, corresponding to the different phases present in the domain, move to track the interface. Despite its accuracy, solver instability is observed with complex interface deformations,

consequently requiring continuous re-meshing at each time step [12, 13]. On the other hand, a less computationally expensive approach is the fixed mesh (figure 1.1- right), in which the grid is stationary. Nonetheless, this approach requires solving a complementary scalar equation [10, 14].

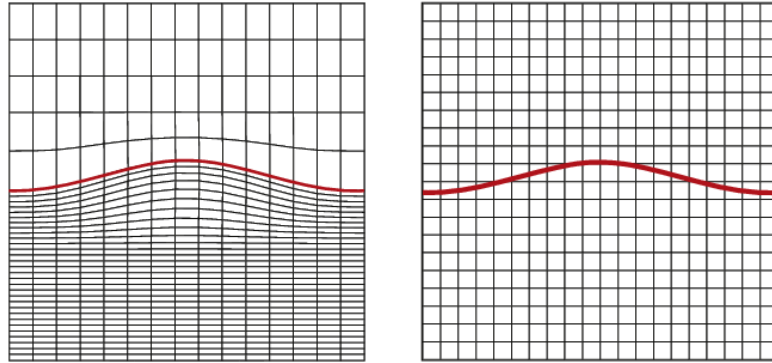


Figure 1.1: Moving mesh (left) versus fixed mesh (right) [15]

Within a Volume of Fluid (VOF) framework, the fixed mesh approach is adopted with the interface implicitly represented by an indicator function, the volume fraction ( $\alpha$ ). It ranges between zero and one; when only two fluids exist in the domain, zero indicates the element is in the bulk of one fluid and one indicates the element is the bulk of the other fluid. Volume fractions between zero and one imply the proximity of the element to the interface.

Although developed earlier [16, 17], the VOF method gained recognition with the SOLA-VOF code of Hirt & Nichols 1981 [18]. It is currently amongst the prevalent methods for multiphase flows [8, 14]; implemented in different commercial and open-source CFD packages, due to its robustness [14], straightforward implementation [19], and mass conservation property [10, 14]. However, the main drawback is lacking an explicit interface equation and the inaccuracies associated with calculating the interface

curvature directly from the VOF function [8, 10, 19], which hinders the inclusion of surface tension forces in a free-surface flow algorithm.

Surface tension forces are often encountered in various applications including but not limited to, material processing [20], enhanced oil recovery [21], and bubble dynamics; from nucleation [22], growth [23], and rising [24] to detachment [25] of bubbles. A comprehensive understanding of the underlying physics and an accurate prediction of such flow phenomena necessitates accounting for surface tension forces. At a microscopic level, these inward-pointing forces arise due to the imbalance between cohesive forces (intermolecular attractive forces) on boundary molecules [26]. Surface tension forces are manifested as a pressure jump across curved interfaces, this concept was introduced by Thomas Young, 1805 [27] and Pierre-Simon Laplace, 1806 [28] and expressed in the Young-Laplace equation [29]:

$$\Delta p = \sigma \kappa \quad (1.1)$$

Where  $\sigma$  is the surface tension coefficient and  $\kappa$  is the radius of curvature.

Successful numerical modeling of surface tension forces has been achieved with, but not limited to, Sharp Surface Tension Force (SSF) [30, 31], Continuum Surface Stress (CSS) [32], and Continuum Surface Force (CSF) [33], with the latter being the first and most extensively employed model within a VOF framework. The CSF method models surface tension forces as volumetric forces acting in interfacial elements. Limitations with the CSF method, however, persist with the generation of spurious currents, also referred to as parasitic currents, in the vicinity of the interface. In extreme cases, these non-physical currents can be enough to break the interface [8], thus adversely influencing the solvers' accuracy and robustness. This highlights the importance of accurate and



consistent implementation of surface tension models both in general and in the CSF method specifically, which is the focus of this work.

Surface tension forces are added to a collocated 2-D incompressible and compressible finite volume-based free surface flow solver on general curvilinear coordinate systems. The code is developed using FORTRAN programming language. Chapter two presents a brief literature on the accurate implementation of the CSF method to achieve force balancing, and the efforts made to reduce spurious current generation through accurate curvature estimation. The governing equations and their general discretization are given in chapter three. The CSF method is presented in chapter four along detailed methodology for curvature estimation using Laplacian Filter, Convolution method, and Height Functions. Different test cases are presented in chapter five and results are compared to the literature to validate the presented formulation. Concluding remarks are drawn in chapter six and recommendations for future work are stated.

## CHAPTER II

### LITERATURE REVIEW

Spurious currents generation can be traced to two main sources. The first is the local numerical imbalances between pressure gradients and surface tension forces [30], which are directly related to the implementation of the surface tension model within the flow algorithm. The second source of spurious currents, inaccurate curvature estimation [34], is attributed to the aforementioned shortcoming of the VOF model.

Extensive numerical studies in the literature attempted to reduce spurious currents by improving curvature estimation [14, 16, 19, 34, 36] within a balanced-force algorithm [8, 37]. Such an algorithm is achieved by careful implementation in which all gradient operators are consistently discretized at the same location [37, 38]. Francois et al., 2005 [30] successfully presented a face-centered formulation in which pressure gradients and the surface tension forces are estimated at cell faces. The surface tension force is then interpolated to the center to be added to the momentum equation. However, spikes in pressure are observed at the interface and limited techniques can be used to compute gradients directly at the face which further poses a problem for unstructured grids. Moukalled et al., 2016 [2] presented a redistributed-cell-centered formulation for the general treatment of body forces in which gradients and forces are calculated at cell centers. In their formulation, a redistributed body force is added to the momentum equation, which when isolated at equilibrium equates to the pressure gradient [39]. Denner et al. [8] along with other recent studies add surface tension directly to cell centers and achieve force balancing by modifying the face-velocity-interpolation procedure only.

Regardless of which formulation is chosen, in a balanced force algorithm body forces including surface tension [49] must be taken into account when interpolating the

advection velocity to the face, otherwise pressure-velocity decoupling might arise [10]. To fully comprehend this, initially consider pressure-velocity coupling alone- neglecting all body forces. Original formulations on collocated grids failed to enforce pressure-velocity coupling due to the errors arising from linear interpolation of velocity from cell centers to faces, where it is needed to compute the mass flux, which in turn led to the checkerboard problem [1, 2]. Solution was then established on staggered grids where velocity fields are stored at cell faces and all other variables, including pressure, are stored at cell centers. This eliminated the need for interpolation hence enforcing pressure-velocity coupling, however, at the cost of large memory and increasing complexity in non-Cartesian grids [9, 40]. Rhie and Chow, 1983 [41] developed an interpolation procedure that allowed the solution on collocated grids, in which a dissipation term is added to the linear interpolation of velocity to cell faces. This term represents the difference between the pressure gradient calculated at the face, using a small stencil and the pressure gradient linearly interpolated to the face, using a large stencil.

Once a balanced-force algorithm is established, the solution is deemed as accurate as the curvature estimation technique used- the second source of spurious currents. Curvature is originally [33] computed from the second spatial derivative of the chosen color function (i.e. volume fraction). Given the abruptly varying nature of the volume fraction field, with interfaces represented over one-cell width only [34], it cannot be directly used to calculate the curvature. Improvement can be noticed by using different equations to calculate the curvature [14, 34, 36] or a more implicit- implementation of the original equation, however, not sufficiently.

One approach to improve curvature estimation is to smooth [42] or map the volume fraction field into a more smoothly-varying-function [34]. Smoothing can be

achieved by applying a Laplacian filter [43, 44, 19], originally proposed by Lafaurie et al. [32]. This is a straightforward method that involves interpolating the VOF to cell faces and back to cell centers prior to curvature estimation. Mapping, on the other hand, can be done by convolving the volume fraction with a smoothing kernel [33, 34, 37, 45, 46], which is a mathematical operation known as convolution. Different parameters are involved in this operation which the literature continuously refines to improve the accuracy of the solution.

Renardy and Renardy [38] employed a least-squares fit of the interface to estimate curvature. Their method is known as the parabolic reconstruction of surface tension or PROST and has been shown to reduce erroneous velocity by two orders of magnitude as compared to the CSF method [50]. PROST was later slightly outperformed with the use of height functions, proposed by Popinet et al. in 2009 [51]. In this approach, the volume fraction field is integrated over a certain domain to reconstruct fluid heights which are then differentiated to obtain curvature. This model has been shown to provide an accurate solution where the interface is well resolved on a fine mesh [8]. Montazeri and Ward [52] later extended this method to non-uniform grids. Other successful methods that involve reconstructing the interface include the piecewise linear interface calculations or PLIC introduced by Youngs [53]. The algorithm was then successfully implemented in `interFoam` by Cifani et al. [54].

More recent efforts in the literature focused on coupling the VOF method with other continuous functions to estimate curvature. Sussman and Puckett [55] exploited the smoothness of the level-set method and proposed a fully coupled level-set-VOF model (CLSVOF). Other couplings between level-set and VOF can be found in the literature such as the simple coupled VOF-level-set method (S-CLSVOF) by Albadawi et al. [56].

In the present work, force balancing is achieved through extension of face velocity interpolation procedure in a novel way. The success of this technique is validated via known benchmark cases (static drop at equilibrium) and compared to the literature. Different curvature estimation techniques are then investigated, mainly Laplacian filter, convolution method, and height functions. The Static bubble and rising bubble cases are then tested to validate curvature estimation.

## CHAPTER III

### GOVERNING EQUATIONS

The main obstacle in solving the momentum and continuity equations, reproduced in equations (3.1) and (3.2) respectively, is the absence of an explicit equation to calculate the pressure field. This is resolved by combining the momentum and continuity equations to form a pressure correction equation. The pressure correction equation is solved after the momentum equation using updated momentum-satisfying mass flow fields. Once the pressure equation is solved, pressure and velocity fields are updated to obtain continuity-satisfying fields. This sequence, which is repeated until convergence, is known as the SIMPLE algorithm (Semi Implicit Method for Pressure Linked Equations), originally developed by Patankar and Spalding [40] on staggered grids and later extended by Rhie and Chow [41] on collocated grids.

$$\frac{\partial}{\partial t}[\rho\mathbf{v}] + \nabla \cdot \{\rho\mathbf{v}\mathbf{v}\} = -\nabla p + \nabla \cdot \{\mu\nabla\mathbf{v}\} + \nabla \cdot \{\mu(\nabla\mathbf{v})^T\} + \mathbf{F}_s \quad (3.1)$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\mathbf{v}) = 0 \quad (3.2)$$

#### 3.1. General Solution on Collocated Grids

Using a first-order Euler scheme for the transient term and a high-resolution scheme for the convection term implemented via the deferred correction approach, the general discretized momentum equation on non-orthogonal grids is:

$$a_C^v \mathbf{v}_C + \sum_{F \sim \text{NB}(C)} a_F^v \mathbf{v}_F = \mathbf{b}_C^v \quad (3.3)$$

$$a_C^v = \frac{\rho_C V_C}{\Delta t} + \sum_{F \sim \text{NB}(C)} \left( \|\dot{m}_f, 0\| + \mu_f \frac{E_f}{d_{CF}} \right) \quad (3.4)$$

$$a_F^v = -\|\dot{m}_f, 0\| - \mu_f \frac{E_f}{d_{CF}} \quad (3.5)$$

$$\begin{aligned} \mathbf{b}_C^v &= \frac{\rho_C^0 V_C}{\Delta t} \mathbf{v}_C^0 + (f_b)_C V_C - \sum_{F \sim \text{NB}(C)} \left( -\mu_f (\nabla \mathbf{v})_f \cdot \mathbf{T}_f + \dot{m}_f (\mathbf{v}_f^{\text{HR}} - \mathbf{v}_f^{\text{U}}) \right) \\ &+ \sum_{F \sim \text{NB}(C)} \left( \mu_f (\nabla \mathbf{v})_f^T \cdot \mathbf{S}_f \right) - \nabla p_C V_C \end{aligned} \quad (3.6)$$

The algebraic form of the pressure correction equation is reproduced below.

$$a_C^{p'} p'_C + \sum_{F \sim \text{NB}(C)} a_F^v \mathbf{v}_F = \mathbf{b}_C^v \quad (3.7)$$

$$a_F^{p'} = -\rho_f D_f \quad (3.8)$$

$$a_C^{p'} = - \sum_{F \sim \text{NB}(C)} a_F^{p'} \quad (3.9)$$

$$\mathbf{b}_C^{p'} = - \sum_{F \sim \text{NB}(C)} \dot{m}_f^* \quad (3.10)$$

Using the minimum correction approach, the  $D_f$  term in Eq. (3.8) can be expressed as

$$D_f = \frac{d_{CF}^x \overline{D}_f^u S_f^x + d_{FC}^y \overline{D}_f^v S_f^y}{(d_{FC}^x)^2 + (d_{FC}^y)^2} \quad (3.11)$$

Which can be written in vector form as

$$D_f = \frac{\mathbf{d}_{FC} \cdot \overline{\mathbf{D}}_f^T \cdot \mathbf{S}_f}{d_{FC}^2} \quad (3.12)$$

The bar accent in the above equation, and in what follows, indicates linear interpolation given in general as follows:

$$\overline{\varphi}_f = (1 - g_F) \varphi_C + g_F \varphi_F \quad (3.13)$$

Where  $g_F$  is a geometric interpolation factor calculated using cell volumes as:

$$g_F = \frac{V_C}{V_C + V_F} \quad (3.14)$$

### 3.2. Volume Fraction Equation

VOF method assumes all phases are immiscible and homogeneous, which allows the solution of only one set of flow equations [30] with varying properties calculated based on the volume fraction as

$$\rho = \sum_{k=1}^n \rho^k \alpha^k \quad (3.15)$$

$$\mu = \sum_{k=1}^n \mu^k \alpha^k \quad (3.16)$$

The volume fraction is calculated by an advection equation, conventionally solved following the momentum and pressure correction equations, derived by Hirt and Nichols [18] as follows.

$$\frac{\partial \rho^k \alpha^k}{\partial t} + \nabla \cdot (\rho^k \mathbf{v} \alpha^k) = 0 \quad (3.17)$$

After discretization, the algebraic equation is written as:

$$a_C^{\alpha^k} \alpha_C^k + \sum_{F \sim \text{NB}(C)} a_F^{\alpha^k} \alpha_F^k = b_C^{\alpha^k} \quad (3.18)$$

$$a_C^{\alpha^k} = \frac{\rho_C^k}{\Delta t} V_C + \sum_{F \sim \text{NB}(C)} \|\dot{m}_f^k, 0\| \quad (3.19)$$

$$a_F^{\alpha^k} = -\|-\dot{m}_f^k, 0\| \quad (3.20)$$

$$b_C^{\alpha^k} = \frac{(\rho_C^k a_C^k)^0}{\Delta t} V_C \quad (3.21)$$



Equations corresponding to  $n-1$  phases is solved and the  $n$ th phase volume fraction is calculated using a geometric constraint Eq. (3.22), in which the summation of volume fraction fields is one.

$$\sum_{k=1}^n \alpha^k = 1 \quad (3.22)$$

### 3.3. Balanced Force Algorithm

As emphasized earlier, in a balanced force algorithm gradient operators must be discretized similarly. Therefore, depending on where gradients are evaluated, two formulations can be established: a face-centered or a cell-centered. Furthermore, surface tension forces should be accounted for in the Rhie-Chow interpolation as mentioned earlier. Regardless of the formulation adopted, the difference between the small stencil and large stencil gradients will be added as a correction to the interpolated face velocity, with different small and large stencils used in different formulations.

#### 3.3.1 Face-Centered Formulation

If pressure gradients are calculated at cell faces, then the volume fraction gradient used to calculate the surface tension force must be directly evaluated at face which leads to a face-centered formulation of the CSF method. Face-centered gradients are calculated using a finite difference scheme. Surface tension forces are then interpolated to cell-centers to be added to momentum equation. The correction term to the Rhie-Chow interpolation would be the difference between body forces evaluated at the face (small stencil) and body forces interpolated from faces to centers and back to cell-faces (large stencil).

### 3.3.2. Redistributed Cell-Centered Formulation

If pressure gradients are calculated at the center, surface tension force should also be evaluated at cell-centers. One can redistribute the surface tension force to retain a stencil similar to that of pressure. The discretized momentum equation is re-written with pressure and body-force terms taken out of source term.

$$a_C^v \mathbf{v}_C + \sum_{F \sim \text{NB}(C)} a_F^v \mathbf{v}_F = \mathbf{b}_C^v - V_C (\nabla p)_C + V_C \overline{\mathbf{B}}_C^v \quad (3.23)$$

The body force is interpolated first to the faces using geometric interpolation.

$$\overline{\mathbf{B}}_f^v = (1 - g_F) \mathbf{B}_C^v + g_F \mathbf{B}_F^v \quad (3.24)$$

Then the face-centered values are averaged at the centers and added to the momentum equation. The twice-averaged, or redistributed body force, should equate to pressure at equilibrium. Thus, considering a static two-phase flow, the momentum equation reduces to

$$0 = -\nabla p_f + \mathbf{B}_f^v \quad (3.25)$$

With  $\nabla p_f$  being a small stencil gradient, the equation above can be re-written as follows.

$$p_F = p_C + \overline{\mathbf{B}}_f^v \cdot \mathbf{d}_{CF} \quad (3.26)$$

Pressure calculated directly at cell center

$$\begin{aligned} \nabla p_C &= \frac{\sum_{f=1}^n P_f \mathbf{S}_f}{V_C} \\ &= \frac{\sum_{f=1}^n \left( (1 - g_F) p_C + g_F (p_C + \overline{\mathbf{B}}_f^v \cdot \mathbf{d}_{CF}) \right) \mathbf{S}_f}{V_C} \\ &= \frac{\sum_{f=1}^n p_C \mathbf{S}_f}{V_C} + \frac{\sum_{f=1}^n g_F (\overline{\mathbf{B}}_f^v \cdot \mathbf{d}_{CF}) \mathbf{S}_f}{V_C} \end{aligned} \quad (3.27)$$

$$\overline{\overline{\mathbf{B}}}_C^v = \frac{\sum_{f=1}^n g_F (\overline{\mathbf{B}}_f^v \cdot \mathbf{d}_{CF}) \mathbf{S}_f}{V_C} \quad (3.28)$$

The advection velocity at the face is then estimated as

$$\mathbf{v}_f = \bar{\mathbf{v}}_f - \bar{\mathbf{D}}_f^v (\nabla p_f^{(n)} - \overline{\nabla p_f^{(n)}}) + \bar{\mathbf{D}}_f^v (\bar{\mathbf{B}}_f^v - \overline{\overline{\mathbf{B}}_f^v}) \quad (3.29)$$

$$\begin{aligned} \dot{\mathbf{m}}_f^* &= \rho \mathbf{v}_f^* \cdot \mathbf{S}_f \\ &= \rho_f \bar{\mathbf{v}}_f^* \cdot \mathbf{S}_f - \rho_f \bar{\mathbf{D}}_f^v (\nabla p_f^{(n)} - \overline{\nabla p_f^{(n)}}) \cdot \mathbf{S}_f \\ &\quad + \rho_f \left[ (\bar{\mathbf{B}}_f^u - \overline{\overline{\mathbf{B}}_f^u}) \cdot \mathbf{S}_f'^x + (\bar{\mathbf{B}}_f^v - \overline{\overline{\mathbf{B}}_f^v}) \cdot \mathbf{S}_f'^y \right] \end{aligned} \quad (3.30)$$

$$\mathbf{S}_f' = \begin{bmatrix} \bar{\mathbf{D}}_f^u & 0 \\ 0 & \bar{\mathbf{D}}_f^v \end{bmatrix} \begin{bmatrix} \mathbf{S}_f^x \\ \mathbf{S}_f^y \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{D}}_f^u \mathbf{S}_f^x \\ \bar{\mathbf{D}}_f^v \mathbf{S}_f^y \end{bmatrix} \quad (3.31)$$

### 3.3.3. Present Formulation

In the presented formulation surface tension is directly added to the momentum equation. Force balancing is achieved through extension of the Rhie-Chow interpolation procedure. The original advection velocity is below.

$$\mathbf{v}_f = \bar{\mathbf{v}}_f - \bar{\mathbf{D}}_f^v (\nabla p_f - \overline{\nabla p_f}) \quad (3.32)$$

As highlighted in earlier sections of this report, achieving force balancing requires consistent discretization of gradients at the same location. Since surface tension term using the CSF (detailed in next chapter) utilizes a large-stencil gradient similar to that of pressure, equation (3.32) can be extended to account for surface tension by adding an analogous term to that of pressure.

$$\mathbf{v}_f = \bar{\mathbf{v}}_f - \bar{\mathbf{D}}_f^v (\nabla p_f - \overline{\nabla p_f}) + \bar{\mathbf{D}}_f^v \sigma K_f (\nabla \alpha_f - \overline{\nabla \alpha_f}) \quad (3.33)$$

$\nabla p_f$  is a corrected small- stencil pressure gradient at the face calculated based on a correction presented in the work of Moukalled et al., 2016 [2] is reproduced in general below:

$$\nabla p_f = \overline{\nabla p_f} + \left[ \frac{p_C - p_F}{d_{FC}} - (\overline{\nabla p_f} \cdot \mathbf{e}_{CF}) \right] \mathbf{e}_{CF} \quad (3.34)$$

$$\nabla \alpha_f = \overline{\nabla \alpha_f} + \left[ \frac{\alpha_C - \alpha_F}{d_{FC}} - (\overline{\nabla \alpha_f} \cdot \mathbf{e}_{CF}) \right] \mathbf{e}_{CF} \quad (3.35)$$

Substituting equations (3.34) and (3.35) in eq. (3.33)

$$\mathbf{v}_f = \overline{\mathbf{v}_f} - \overline{\mathbf{D}_f^v} \left[ \frac{p_F - p_C}{d_{CF}} - (\overline{\nabla p_f} \cdot \mathbf{e}_{CF}) \right] \mathbf{e}_{CF} \quad (3.36)$$

$$+ \overline{\mathbf{D}_f^v} \sigma K_f \left[ \frac{\alpha_C - \alpha_F}{d_{FC}} - (\overline{\nabla \alpha_f} \cdot \mathbf{e}_{CF}) \right] \mathbf{e}_{CF}$$

$$\mathbf{e}_{CF} = \frac{\mathbf{d}_{CF}}{d_{CF}} \quad (3.37)$$

Where

$$\mathbf{d}_{CF} = \mathbf{r}_F - \mathbf{r}_C \quad (3.38)$$

Keeping in mind that  $\mathbf{d}_{CF} = -\mathbf{d}_{FC}$ , replacing Eq. (3.37) in (3.36) and taking  $1/d_{CF}$  in common, the extended face-velocity equation is expressed as:

$$\begin{aligned} \mathbf{v}_f = \overline{\mathbf{v}_f} - \overline{\mathbf{D}_f^v} \left[ (p_F - p_C) - (\overline{\nabla p_f} \cdot \mathbf{d}_{CF}) \right] \frac{\mathbf{d}_{CF}}{d_{CF}^2} \\ + \overline{\mathbf{D}_f^v} \sigma \left[ \kappa_f (\alpha_F - \alpha_C) - (\kappa_f \overline{\nabla \alpha_f} \cdot \mathbf{d}_{CF}) \right] \frac{\mathbf{d}_{CF}}{d_{CF}^2} \end{aligned} \quad (3.39)$$

The mass flow at the face can be then computed as follows

$$\begin{aligned} \dot{m}_f^* &= \rho_f \mathbf{v}_f^* \cdot \mathbf{S}_f \\ &= \rho_f \overline{\mathbf{v}_f^*} \cdot \mathbf{S}_f - \rho_f \left[ (p_F - p_C) - (\overline{\nabla p_f} \cdot \mathbf{d}_{CF}) \right] \frac{\mathbf{d}_{CF} \cdot \overline{\mathbf{D}_f^v}^T \cdot \mathbf{S}_f}{d_{CF}^2} \end{aligned} \quad (3.40)$$

$$+ \rho_f \overline{\mathbf{D}_f^v} \sigma \left[ \kappa_f (\alpha_F - \alpha_C) - (\kappa_f \overline{\nabla \alpha_f} \cdot \mathbf{d}_{CF}) \right] \frac{\mathbf{d}_{CF} \cdot \overline{\mathbf{D}_f^v}^T \cdot \mathbf{S}_f}{d_{CF}^2}$$

$$\begin{aligned} \Rightarrow \dot{m}_f^* &= \rho_f \overline{\mathbf{v}_f^*} \cdot \mathbf{S}_f - \rho_f D_f \left[ (p_F - p_C) - (\overline{\nabla p_f} \cdot \mathbf{d}_{FC}) \right] \\ &+ \rho_f \sigma D_f \left[ \kappa_f (\alpha_F - \alpha_C) - (\kappa_f \overline{\nabla \alpha_f} \cdot \mathbf{d}_{CF}) \right] \end{aligned} \quad (3.41)$$

Equation (3.39) can be found in the work of Denner et al. [8] along with further extension to account for gravitational forces and transient effects.

## CHAPTER IV

### SURFACE TENSION AND CURVATURE MODEL

#### 4.1. CSF Model

Using the Continuum Surface Force model [33], the surface tension force is approximated as:

$$F_{\text{CSF}} = f_s \delta_s \quad (4.1)$$

Where  $\delta_s$  is a surface delta function and  $f_s$  is the surface tension force per unit interfacial area given as follows.

$$f_s = \sigma \kappa \hat{\mathbf{n}} + \nabla_s \sigma \quad (4.2)$$

$\sigma$  is the surface tension coefficient,  $\nabla_s$  is the surface gradient,  $\kappa$  is the interfacial curvature, and  $\hat{\mathbf{n}}$  is the interface unit normal given as

$$\mathbf{n} = \nabla \alpha \quad (4.3)$$

$$\hat{\mathbf{n}} = \frac{\nabla \alpha}{|\nabla \alpha|} \quad (4.4)$$

The first term in Eq. (4.2) is the normal component while the second is the tangential. Assuming a constant surface tension coefficient, the tangential component can be neglected. The surface delta function is taken to be as the magnitude of the normal vector as to be non-zero only in transition regions. The volumetric force is thus re-written as:

$$F_s = \sigma \kappa \nabla \alpha \quad (4.5)$$

Density scaling has been conventionally used to reduce spurious currents at large density ratios. The density scaled CSF is given by:

$$F_{\text{CSF}} = \frac{\rho}{\rho_{\text{avg}}} F_{\text{CSF}} \quad (4.6)$$

## 4.2. Curvature Equations

Original curvature equation proposed by Brackbill et al., 1992 [33] estimated the curvature as the divergence of the unit normal vector:

$$\kappa = -\nabla \cdot \hat{\mathbf{n}} \quad (4.7)$$

Eq. (4.7) can be extended in which the unit normal vector is accounted for implicitly as follows:

$$\begin{aligned} \kappa &= -\frac{\partial}{\partial x} \left( \frac{\partial \alpha}{\partial x} / |\mathbf{n}| \right) - \frac{\partial}{\partial y} \left( \frac{\partial \alpha}{\partial y} / |\mathbf{n}| \right) \\ &= -\frac{1}{|\mathbf{n}|^2} \left( \frac{\partial^2 \alpha}{\partial x^2} |\mathbf{n}| - \frac{\partial \alpha}{\partial x} \frac{\partial |\mathbf{n}|}{\partial x} + \frac{\partial^2 \alpha}{\partial y^2} |\mathbf{n}| - \frac{\partial \alpha}{\partial y} \frac{\partial |\mathbf{n}|}{\partial y} \right) \end{aligned} \quad (4.8)$$

Eq. (4.8) can be written in the following format, which is proposed by Rudman, et al. 1998 [36]

$$\kappa = \frac{1}{|\mathbf{n}|} \left( \frac{1}{|\mathbf{n}|} \frac{\partial \alpha}{\partial x} \frac{\partial |\mathbf{n}|}{\partial x} - \frac{\partial^2 \alpha}{\partial x^2} - \frac{\partial^2 \alpha}{\partial y^2} + \frac{1}{|\mathbf{n}|} \frac{\partial \alpha}{\partial y} \frac{\partial |\mathbf{n}|}{\partial y} \right) \quad (4.9)$$

Further extension of the equation can be made where:

$$|\mathbf{n}| = \sqrt{\left( \frac{\partial \alpha}{\partial x} \right)^2 + \left( \frac{\partial \alpha}{\partial y} \right)^2} \quad (4.10)$$

$$\frac{\partial |\mathbf{n}|}{\partial x} = \frac{1}{|\mathbf{n}|} \left( \frac{\partial^2 \alpha}{\partial x^2} \frac{\partial \alpha}{\partial x} + \frac{\partial^2 \alpha}{\partial x \partial y} \frac{\partial \alpha}{\partial y} \right) \quad (4.11)$$

$$\frac{\partial |\mathbf{n}|}{\partial y} = \frac{1}{|\mathbf{n}|} \left( \frac{\partial^2 \alpha}{\partial y \partial x} \frac{\partial \alpha}{\partial x} + \frac{\partial^2 \alpha}{\partial y^2} \frac{\partial \alpha}{\partial y} \right) \quad (4.12)$$

Substituting equations (4.10), (4.11), and (4.12) in (4.9) and with further manipulation, the curvature equation used in this work is given in Eq. (4.13).

$$\kappa = \frac{1}{\left[ \left( \frac{\partial \alpha}{\partial x} \right)^2 + \left( \frac{\partial \alpha}{\partial y} \right)^2 \right]^{3/2}} \left[ \frac{\partial^2 \alpha}{\partial x^2} \left( \frac{\partial \alpha}{\partial y} \right)^2 - \frac{\partial \alpha}{\partial x} \frac{\partial \alpha}{\partial y} \left( \frac{\partial^2 \alpha}{\partial x \partial y} + \frac{\partial^2 \alpha}{\partial y \partial x} \right) + \frac{\partial^2 \alpha}{\partial y^2} \left( \frac{\partial \alpha}{\partial x} \right)^2 \right] \quad (4.13)$$

### 4.3. Laplacian Filter

Lafaurie et al. [32] and Ubbink [42] introduced the Laplacian Filter as an iterative method to smooth the volume fraction field that will be used to calculate the radius of curvature. Its implementation is straightforward due to its discrete nature. The smoothed field can be calculated using equation (4.14), where  $f$  is the index that runs through all the faces of cell( $i,j$ ), shown in figure 4.1, and  $S_f$  is the corresponding surface vector. The values at the faces are calculated via linear interpolation.

$$\tilde{\alpha} = \frac{\sum_{f=1}^n \bar{\alpha}_f S_f}{\sum_{f=1}^n S_f} \quad (4.14)$$

As such, only the adjacent neighbor cells are used to approximate the value of the central cell [8]. By applying this filter iteratively, the effect of second-order neighbors or more can be accounted for. It should be emphasized, however, that smoothing diffuses the interface and should thus be used only to the extent that is required to adequately suppress spurious currents [44]. Ubbink [42] suggested applying the filter at least twice, the contribution of the neighboring cells would thus be 5/16, 1/8, & 3/128 for central cell, first order neighbors, and second order neighbors respectively.



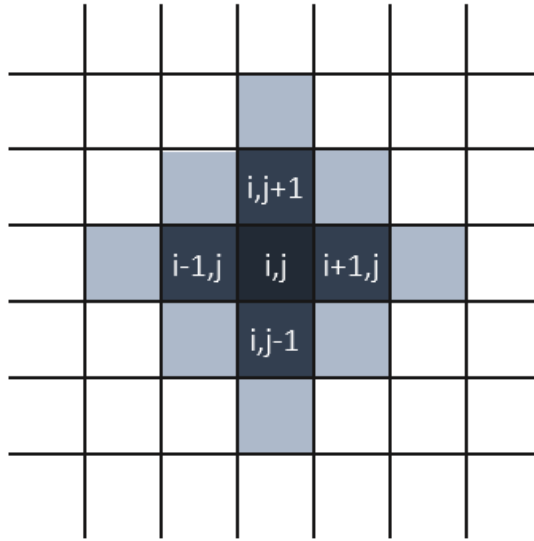


Figure 4.1: Neighboring cells used in Laplacian Filter

#### 4.4. Convolution

Convolution is a mathematical operation in which two functions are merged, over a certain domain, to form a third function. The latter embodies the effect of one of the original functions, usually a convolution kernel, on the other function under study.

Convolution is amongst the common methods used to estimate radius of curvature and reduce spurious currents. The kernel would thus act as a weight to smooth the VOF field within the interfacial region. The domain over which the color function is convolved is referred to as support length  $d$ .

The smoothed volume fraction ( $\tilde{\alpha}$ ) and normal vector are expressed as

$$\hat{\mathbf{n}}_c = (K_x * \alpha, K_y * \alpha) \quad (4.15)$$

$$\tilde{\alpha}(\mathbf{x}) = K_x * \alpha = \int \alpha(\check{\mathbf{x}})K(d - \check{\mathbf{x}})d\check{\mathbf{x}} \quad (4.16)$$

#### 4.4.1 Choice of Kernel

The choice of kernel is critical as it directly influences the effectiveness of this technique. Different convolution kernels used in this scope of work can be found in the literature, such as polynomial [46], spline [33], and trigonometric functions [17]. When selecting a convolution kernel  $K$ , however, it is necessary to consider the following conditions outlined by Williams et al. [46]:

1. Compact support: the kernel function should be non-zero over the support length and zero elsewhere.
2. Symmetry: the kernel function should be symmetric or have an odd symmetry about the center.
3. Monotonicity: the kernel function should decrease in magnitude as the distance from the center of the kernel increases.
4. Smoothness: the kernel function should be continuous and smooth, with sufficient differentiability to avoid introducing numerical errors.
5. Normalization: the integral of the kernel function over its entire domain should be equal to one.

Williams [46] comprehensive study on convolution kernels showed that sixth- and eighth-order polynomial kernels outperformed spline kernels. The kernels most used in the literature, including the sixth and eighth-order kernels, are given as:

$$K_6(r, d) = \begin{cases} A(d^2 - r^2)^3 & \text{if } \frac{r}{d} < 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.17)$$

$$K_8(r, d) = \begin{cases} A(d^2 - r^2)^4 & \text{if } \frac{r}{d} < 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.18)$$

$$K_2(r, d) = \begin{cases} A(1 - r/d)^4 \cdot (1 + 4r/d) & \text{if } \frac{r}{d} < 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.19)$$

$$K_4(r, d) = \begin{cases} A(1 - r/d)^6 \cdot (3 + 18r/d + 35(r/d)^2) & \text{if } \frac{r}{d} < 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.20)$$

$$K_{\text{Cubic-Spline}}(r, d) = \begin{cases} A(1 - r/d)^3 \cdot (1 + 3r/d) & \text{if } \frac{r}{d} < 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.21)$$

$$K_{\text{B-Spline}}(r, d) = \begin{cases} A(1 - r/d)^4 \cdot [1 + 4r/d + (16/3)(r/d)^2] \cdot (1/2) & \text{if } \frac{r}{d} < 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.22)$$

$$K_{\text{Quintic}}(r, d) = \begin{cases} A(1 - r/d)^5 \cdot (1 + 5r/d + 5(r/d)^2 + 8(r/d)^3) & \text{if } \frac{r}{d} < 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.23)$$

$$K_{\text{Quintic B-Spline}}(r, d) = \begin{cases} A(1 - r/d)^5 \cdot (1 + 5r/d + 20(r/d)^2) & \text{if } \frac{r}{d} < 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.24)$$

$$K_{\text{Gaussian}}(r, d) = \begin{cases} Ae^{-(r/d)^2} & \text{if } \frac{r}{d} < 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.25)$$

Where  $d$  is the smoothing length,  $r$  is the distance to kernel's center, and  $A$  is a normalizing constant. Peskin [17] introduced a cosine function as a kernel given in the following equation:

$$K_{\text{cos}}(x, d) = \begin{cases} \left( (A/2d)(1 + \cos(\frac{\pi X}{d})) \right) & \text{if } \frac{X}{d} < 1 \\ 0 & \text{otherwise} \end{cases} \quad (4.26)$$

$$\check{\alpha}(x) = \int \alpha(\check{x})K_{\text{cos}}(\check{x} - x)K_{\text{cos}}(\check{y} - y)d\check{x} \quad (4.27)$$

#### 4.4.2. Normalization Factor

The normalizing factor in these kernels is used to ensure that the sum of the kernel coefficients is equal to one. In a discrete formulation, the kernel is typically implemented

as a finite set of coefficients that are designated to neighboring cells. The normalizing factor would then be calculated as the inverse of the sum of those coefficients [30].

To gain deeper insights into the impact of various kernels on the VOF field, figure 4.2 offers valuable visualization and comparison of the weight distribution exhibited by different kernels as a function of the distance from the center of the kernel.

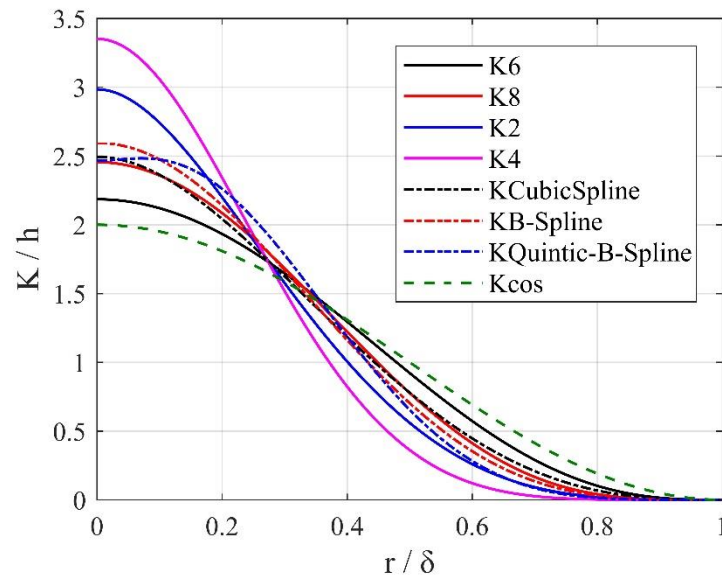


Figure 4.2: Kernel weight distribution as a function of distance from its center

#### 4.4.3. Support Length

It is desired to limit the values of the kernel derivative to a minimum. Therefore, as Cummins et al. [34] inferred, the kernel must be resolved within a wide stencil. Francois et al. [30] investigated the effect of different support lengths for polynomial kernels, both constant kernel support and function of grid spacing. The study found that convergence can be achieved with support length of constant support length of  $d=0.4$ , which corresponds to  $R/5$ - independent of grid spacing. Furthermore, the solution with a support length of  $d=3h$  diverged, which agrees with the findings of Williams et al. [46]

and Cummins [34] who suggested that the support length should extend over at least four elements in each direction ( $d=9h$ , where  $h$  is the grid spacing).

## 4.5. Height Function

The use of height functions for estimating curvature has become one of the standard techniques and has been further developed and refined by researchers in the field. The procedure for evaluating curvature using HF, as presented in the work of [57], is detailed below.

### 4.5.1. Identify the Interfacial Area

It is better to evaluate curvature in interfacial cells only to minimize numerical errors. As such the procedure starts with identifying which elements have a nonzero VOF gradient, which literature [58] suggests is suitable to represent the interface. This can be achieved by applying the inequality in Eq. (4.284.2), where  $h$  is the grid spacing and  $\epsilon_1$  is a positive constant set to  $10^{-6}$  based on the work of [57]

$$|\nabla \cdot \alpha| > \frac{\epsilon_1}{h} \quad (4.28)$$

### 4.5.2. Extend Computational Stencil

Originally, curvature calculations using HF were made by applying finite difference scheme at the cell-center using a 3x3 stencil [59]. Research then showed that Height Functions require larger stencils to yield accurate results. Cummins et al. [34] suggested extending to a 7x3 or 3x7 stencil, where applicable, to resolve the interface. Hernández et al. [60] further enhanced the implementation by making effective

modifications and limitations to the stencil extension. Conditions for stencil extension in vertical direction are detailed below, where subscript  $j+u$  represents the  $u^{\text{th}}$  row above the cell under study and  $j-d$  represents the  $d^{\text{th}}$  row below that cell. Similar criteria for horizontal extension can be applied.

Upward extension:

$$\text{Sin}(\mathbf{n}_y) \cdot \sum_{s=-1}^{s=1} (\alpha_{i+s,j+u} - \alpha_{i+s,j+u-1}) > 0 \quad (4.29)$$

$$\epsilon_1 < \sum_{s=-1}^{s=1} \alpha_{i+s,j+u} > 3 - \epsilon_1 \quad (4.30)$$

Downward extension:

$$\text{Sin}(\mathbf{n}_y) \cdot \sum_{s=-1}^{s=1} (\alpha_{i+s,j-d} - \alpha_{i+s,j-d+1}) > 0 \quad (4.31)$$

$$\epsilon_1 < \sum_{s=-1}^{s=1} \alpha_{i+s,j-d} > 3 - \epsilon_1 \quad (4.32)$$

Although one direction is required to calculate the derivative and estimate curvature, initially the extension is performed in both directions to determine which direction produces the most accurate results.

#### 4.5.3. Correct VOF in Extended Stencil

Once extension in both directions is achieved, a correctional step must be implemented before choosing the actual height function direction, to avoid errors when multiple interfaces cut the same stencil. Therefore, one must check if any cell reverses monotonicity of the volume fraction and if so, set it's VOF to zero or one, depending on the following criteria:

Cells above the  $j^{\text{th}}$  row will be corrected if

$$\text{Sign}(\mathbf{n}_y) \cdot (\alpha_{i+s,j+u} - \alpha_{i+s,j+u-1}) < \epsilon_2 \quad ; \quad s = -1, 0, 1 \quad (4.33)$$

Then the volume fraction of that cell is set to

$$\alpha_{i+s,j+u} = \frac{1}{2} \cdot (1 + \text{Sign}(\mathbf{n}_y)) \quad (4.34)$$

Cells above the  $j^{\text{th}}$  row will be corrected if

$$\text{Sign}(\mathbf{n}_y) \cdot (\alpha_{i+s,j-d} - \alpha_{i+s,j-d+1}) > \epsilon_2 \quad s = -1, 0, 1 \quad (4.35)$$

Then the volume fraction of that cell is set to

$$\alpha_{i+s,j-d} = \frac{1}{2} \cdot (1 - \text{Sign}(\mathbf{n}_y)) \quad (4.36)$$

where  $\epsilon_2$  is a positive constant set to  $10^{-3}$  based on the work of [57].

This modification ensures that the volume fraction values in each column increase or decrease consistently along the extension direction. As such the errors arising from the other curves that are not currently under study are mitigated.

#### ***4.5.4. Choose Height Function Direction***

From an analytical perspective, curvature can be estimated using HF extended in either direction. On the other hand, Popinet [51] and Magnini [61] showed that the numerical errors increase as a function of the first derivative of the HF. As such, for any given cell in the domain, the direction that yields the lowest magnitude of the HF first derivative shall be used.

If the inequality below holds true, then the horizontally computed first-derivative is smaller in magnitude, which implies that vertical extension shall be employed. The opposite is true; if this inequality is false, horizontal extension is employed.

$$\left| \sum_{s=-1}^{s=r} (\alpha_{i+s,j+1} - \alpha_{i+s,j-1}) \right| \geq \left| \sum_{s=-u}^{s=d} (\alpha_{i+1,j+s} - \alpha_{i-1,j+s}) \right| \quad (4.37)$$

Where  $i+r$  denotes the  $r^{\text{th}}$  cell on the right and  $i-1$  denotes the  $1^{\text{th}}$  cell on the left.

#### 4.5.5. Calculate Height Function and Curvature

Once the extension direction has been chosen, the height functions of each column (or row) can be calculated. Assuming vertical extension was chosen in the, the height function for each column can be calculated as follows:

$$H_{i+s} = \sum_{t=-d}^u \alpha_{i+s,j+t} \cdot \Delta y, s = -1,0,1 \quad (4.38)$$

The derivatives and consequently curvature can then be calculated by applying second order finite difference schemes:

$$H_x = \frac{H_{i+1} - H_{i-1}}{2\Delta x} \quad (4.39)$$

$$H_{xx} = \frac{H_{i+1} - 2H_i + H_{i-1}}{\Delta x^2} \quad (4.40)$$

$$\kappa_{i,j} = \frac{H_{xx}}{(1 + H_x^2)^{3/2}} \quad (4.41)$$

#### 4.5.6. Distribute Curvature

Some of the cells identified as belonging to the interface, using Eq. (4.284.2) in section 4.5.1, may rather be non-interfacial, and in reality, fall exactly next to an interfacial cell; thus, having a non-zero gradient. This problem has been highlighted in the literature [34, 60, 61] and can be overcome by setting the curvature in these cells to



the nearest real curvature as proposed by Magnini [61] in his explicit procedure as follows:

$$\text{offset}_{i,j} = \begin{cases} u - \text{floor}\left(\frac{H_i}{\Delta y}\right) & \text{if } \text{sign}(n_y) > 0 \\ \text{floor}\left(\frac{H_i}{\Delta y}\right) - d & \text{if } \text{sign}(n_y) < 0 \end{cases} \quad (4.42)$$

A non-zero offset value implies that the height falls within a neighboring cell. Consequently, Eq. (4.43) is used to set the curvature of the cell equal to the curvature of that nearby interfacial cell.

$$\kappa_{i,j} = \kappa_{i,j-\text{offset}} \quad (4.43)$$

#### **4.5.7. Bounding Height Function**

Section 4.5.1 presented above illustrates one of the methods generally used to identify interfacial elements and 4.5.6 follows as a necessary corrective measure. Alternatively, one can identify interfacial elements using an approach adapted more to height functions, suggested in the work of Cummins et al. [34]. In this method, the curvature is evaluated only if the height function is within the cell boundaries. As such, sections 4.5.1 and 4.5.6 presented earlier are dropped, and once the height function is obtained, the curvature is calculated iff:

$$Y_{j-1} < H_{i,j} < Y_{j+1} \quad (4.44)$$

# CHAPTER V

## NUMERICAL TEST CASES

### 5.1. Inviscid Stationary Fluid Column

Prior to investigating surface tension, force balancing between pressure gradients and gravitational forces must be ensured. For this purpose, a 2D inviscid stationary fluid column is considered. In this test case all forces are neglected except gravity, which is specified as  $9.8\text{m/s}^2$  acting in downward direction. Velocity and pressure fields are initialized with zero values within a  $1 \times 1$  square domain, as shown in figure 5.1. Initially, this domain is half-filled with the heavy fluid (density =  $\rho_h$ ) and the density ratio is set to  $\rho_h/\rho_l = 10^{+6}$ .

The exact pressure distribution can be found analytically as follows:

$$P_e = \begin{cases} \rho_l |g| (1 - y) & 0.5 \leq y \leq 1 \\ 0.5 * \rho_l |g| + \rho_h |g| (0.5 - y) & 0 \leq y < 0.5 \end{cases} \quad (5.1)$$

Errors in pressure and velocity are of interest to quantify numerical imbalances and are calculated as follows:

$$L_2(P) = \sqrt{\frac{\sum_{i=1}^N (|P_i - P_e|^2 V_i)}{\sum_{i=1}^N |P_e|^2 V_i}} \quad (5.2)$$

$$L_\infty(\mathbf{V}) = \max(\|\mathbf{V}\|) \quad (5.3)$$

Simulations were conducted for one and one hundred timesteps of  $\Delta t = 10^{-4}$ , following the work of Huang et al. [49]. Figure 5.2 illustrates the maximum velocity magnitude in the domain  $L_\infty(\mathbf{V})$  as a function of grid spacing. The velocity magnitude agrees with the findings of [49] over fine grids ( $\Delta x = \Delta y = 1/160$ ), however, slightly deviates for courser grids. For the coarsest mesh ( $\Delta x = \Delta y = 1/20$ ) the velocity magnitude after one

timestep is  $9.22\text{E-}14$ , which is lower than  $1.14\text{E-}11$  reported in [49]. However, errors in velocity after one hundred timesteps amount to  $1.1\text{E-}8$  which is one order of magnitude larger than that of [49].

On the other hand, errors in pressure  $L_2$  (P), as presented in Table 1, were found to be of order  $O(10^{-9})$  and  $O(10^{-8})$  after one and one hundred timestep, respectively. Both of which are higher than pressure errors reported in [49], yet present results still fall within acceptable range and are of the same order of magnitude of the specified tolerance.

Therefore, this algorithm is capable of balancing gravitational forces and pressure gradients, while maintaining errors at solver's tolerance.

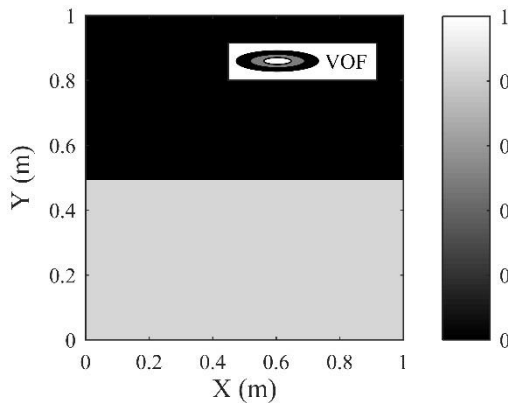


Figure 5.1: Stationary fluid column volume fraction field.

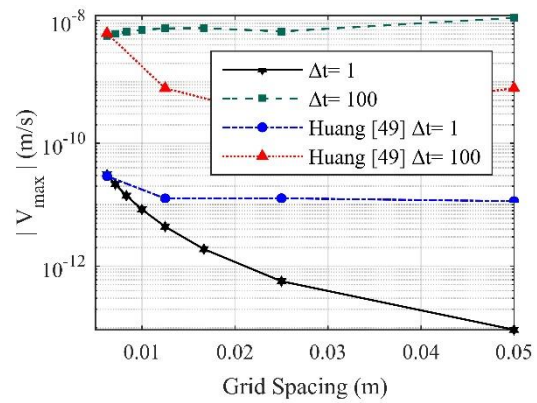


Figure 5.2: Stationary fluid column maximum velocity as a function of grid ratio.

Table 1: Errors in calculated pressure as a function of grid spacing after one and hundredth timestep.

$\Delta x = \Delta y$	$\Delta t = 1$		$\Delta t = 100$	
	$L_2(P)$	Huang [49]	$L_2(P)$	Huang [49]
0.05	9.22E-11	4.16E-12	6.13E-08	7.61E-12
0.025	3.73E-10	1.55E-12	5.99E-08	7.00E-12
0.01667	8.34E-10	-	6.06E-08	-
0.0125	1.47E-09	2.29E-12	6.16E-08	4.94E-12
0.01	2.27E-09	-	6.25E-08	-
0.00833	3.20E-09	-	6.34E-08	-
0.00714	4.33E-09	-	6.43E-08	-
0.00625	5.55E-09	1.10E-12	6.51E-08	3.92E-12

## 5.2. Inviscid Static Drop at Equilibrium

The case of an inviscid static drop at equilibrium is one of the commonly used benchmark cases to test force balancing, curvature calculation, and quantify spurious current generation. All body forces except surface tension are neglected. Following the work of Francois et al., 2005 [30], three pressure jumps are considered:

$$\Delta P_{\text{total}} = P_{\text{in}} - P_{\text{out}}$$

$$P_{\text{in}} = \text{average pressure inside drop, } r < R$$

$$P_{\text{out}} = \text{average outside the drop, } r > R$$

$$\Delta P_{\text{partial}} = P_{\text{in}} - P_{\text{out}}$$

$$P_{\text{in}} = \text{average pressure inside drop, } r < R/2$$

$$P_{\text{out}} = \text{average outside the drop, } r > 3R/2$$

$$\Delta P_{\text{max}} = P_{\text{max}} - P_{\text{min}}$$

$$P_{\text{min}} = \text{min pressure in domain}$$

$$P_{\text{max}} = \text{max pressure in domain}$$

The relative pressure jump error is calculated as:

$$E(\Delta p_n) = \frac{|\Delta p_n - \Delta p_{\text{exact}}|}{\Delta p_{\text{exact}}} \quad (5.4)$$

Where  $n$  denotes the different pressure jumps discussed, total, partial and max. It is worth mentioning that  $\Delta P_{\text{partial}}$  only measure the pressure jump in the bulk of the fluids, unlike  $\Delta P_{\text{total}}$  which also reflects the transition region.

### 5.2.1. Exact Curvature

Initially, an exact curvature is specified which is  $1/R_{\text{drop}}$  in a two-dimensional domain. The exact pressure jump across the interface is given in the Young-Laplace's equation. This ensures that errors in pressure jump across the drop and velocity magnitudes larger than the solver tolerance are solely due to numerical imbalances.

The drop of radius  $R = 2m$  is centered in an  $8 * 8$  domain and the mesh is  $40 * 40$  ( $R/h = 10$ ) elements. Velocity and pressure fields are initialized with zeros and boundary conditions are set to walls. Fluid density inside the drop ( $\rho_1$ ) is 1 and the fluid density outside ( $\rho_2$ ) is allowed to vary. Using a convergence criterion of  $10^{-15}$ , different implementations of the CSF method are compared at different density ratios. Results for the maximum velocity magnitude and pressure jump after one time step of  $10^{-6}$  are shown in Table 2. The redistributed CSF, redistributed and density scaled, are compared to the face centered of Francois et al., 2005 [30] and to the cell centered proposed earlier. The CSF method performance is then compared to the SSF method of [30].

Table 2: The maximum velocity magnitude and pressure jump after one time step of  $10^{-6}$

	$\rho_1/\rho_2$	$V_{\text{mag max}}$	$E(\text{DP})_{\text{total}}$	$E(\text{DP})_{\text{partial}}$	$E(\text{DP})_{\text{max}}$
<b>Redistributed CSF</b>	1	5.71E-05	5.44E-02	2.25E-04	1.19E-01
	$10^{+3}$	4.90E-02	8.36E-02	3.33E-02	1.61E-01
	$10^{+5}$	4.49	1.01E-01	5.22E-02	1.32E-01
<b>Redistributed-density-scaled CSF</b>	1	5.71E-05	5.44E-02	2.25E-04	1.19E-01
	$10^{+3}$	4.77E-02	9.68E-02	1.17E-02	2.40E-01
	$10^{+5}$	3.88	1.00E-01	2.06E-02	2.00E-01
<i>Density-scaled CSF [30]</i>	1	2.28E-05	7.76E-02	1.03E-14	7.79E-14
	$10^{+3}$	1.33E-04	5.6E-01	5.78E-01	4.40E-01
	$10^{+5}$	1.33E-04	5.62E-01	5.81E-01	4.42E-01
<i>Face-centered CSF [30]</i>	1	1.25E-18	2.89E-2	2.73E-15	8.72E-14
	$10^{+3}$	4.97E-18	2.89E-2	1.95E-15	1.36E-14
	$10^{+5}$	5.70E-19	2.89E-2	7.79E-16	1.38E-14
<b>Cell-centered CSF</b>	1	3.72E-20	0	0	1.94E-16
	$10^{+3}$	5.56E-17	1.94E-15	5.84E-16	5.40E-15
	$10^{+5}$	5.62E-15	1.94E-15	5.84E-16	5.45E-15
<i>Face-centered SSF [30]</i>	1	5.43E-19	1.36E-15	5.84E-15	1.63E-14
	$10^{+3}$	4.44E-18	1.95E-16	1.17E-15	3.11E-15
	$10^{+5}$	2.71E-19	3.89E-16	3.7E-15	4.87E-15

The redistributed CSF shows generation of non-physical flows at high density ratios with errors in the pressure jump of  $O(10^{-2})$ . Spurious currents are reduced with density scaling as reported by Francois et al. [30]. On the other hand, the face-centered [30] shows better performance with maximum velocity magnitude reducing to solver

tolerance. However, the total pressure is limited to  $O(10^{-2})$  indicating the presence of small spikes in pressure within the drop. This is resolved in this cell-centered implementation of CSF which shows superior results to the other formulations and comparable results to the SSF method of [30].

The pressure distribution is shown below (figures 5.3) for the redistributed and cell-centered implementations, with the latter presenting exact pressure jump, unlike the other.

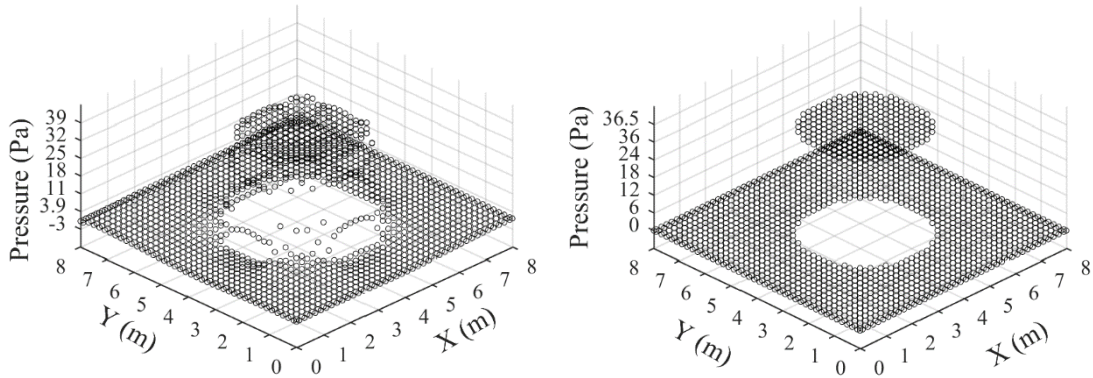


Figure 5.3: Computed pressure with different CSF models for the inviscid static drop case when the exact curvature is specified. The grid is uniform and of resolution  $R/h = 10$ . The density ratio is 103. Redistributed left and cell centered right.

The performance of both techniques, cell centered and redistributed, has been further investigated at higher density ratios and for different grid spacings. The figures below illustrate the error in pressure jump (total and partial) and velocity magnitude (maximum and average) as a function of density ratio (figures 5.4 and 5.5) as well as grid spacing (figures 5.6 and 5.7) for both cell-centered and redistributed methods.

It is important to note that the solution diverged with the redistributed method for high density ratios ( $>10^5$ ), unlike the cell-centered method. However, both techniques exhibited consistent performance across various density ratios and grid spacings, except

for an increase in spurious current generation at high density ratios, which nevertheless remained within an acceptable range.

Figures 5.8 and 5.9 compare the maximum pressure jump and maximum velocity, respectively, to the findings of Huang et al. [49]. In their work, a force balanced algorithm is tested on unstructured Delaunay triangular grids with the same grid spacing but approximately double the number of elements. For this comparison the time step has been reduced to  $10^{-4}$  and density ratio is set to  $10^3$ . The results indicate that the cell-centered method achieves better outcomes for coarse meshes in terms of the maximum pressure jump, while producing comparable results for finer meshes. Thus, the cell centered method restores exact force balancing between pressure and surface tension forces for different mesh spacing and density ratios; with negligible spurious current generation and exact pressure jump restored.

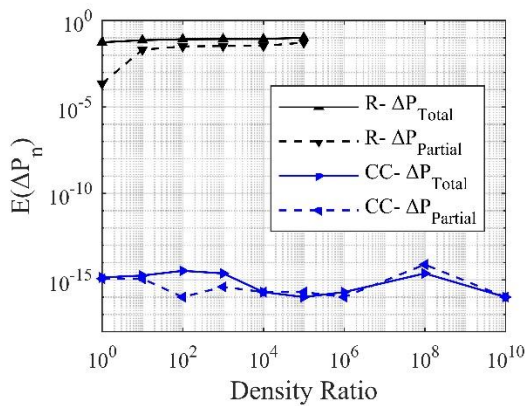


Figure 5.4: Errors in calculated pressure as a function of density ratio.

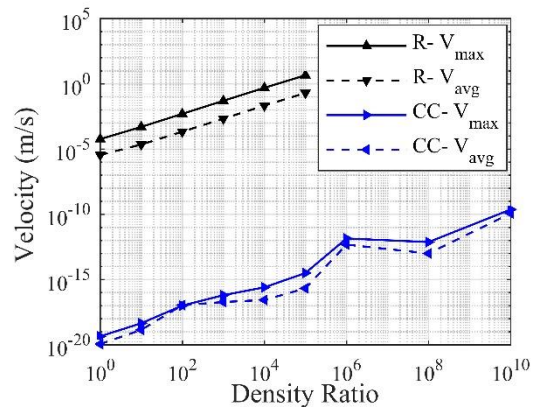


Figure 5.5: Errors in velocity magnitude as a function of grid spacing.



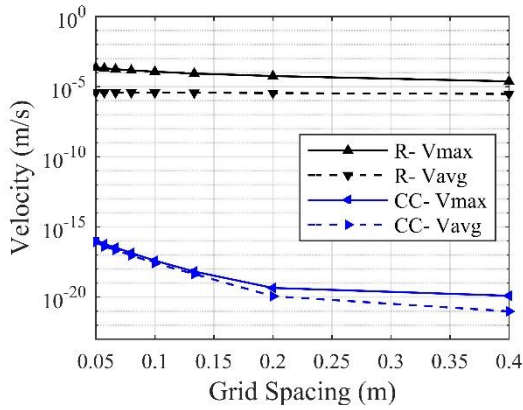


Figure 5.6: Errors in velocity magnitude as a function of grid spacing.

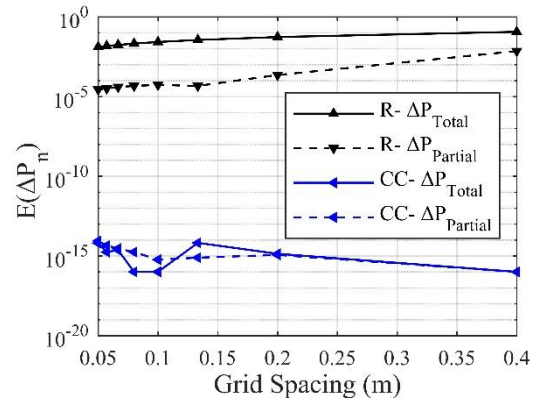


Figure 5.7: Errors in calculated pressure as a function of grid spacing.

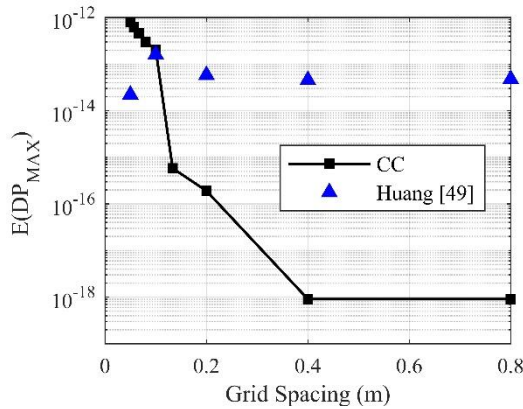


Figure 5.8: Error in calculated pressure as a function of grid spacing.

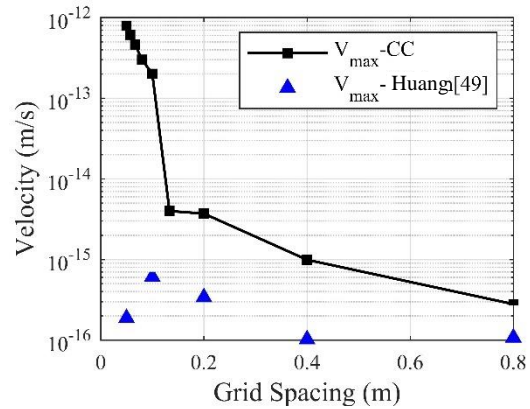


Figure 5.9: Error in maximum velocity as a function of grid spacing.

The figure below (5.) show maximum residuals obtained with different CSF models for the same inviscid static drop case when the exact curvature is specified. The grid is uniform and of resolution  $R/h = 10$ . The surface tension coefficient is 73 and the density ratio is 1. The lowest residuals- same as solvers tolerance, are achieved with the cell centered method.

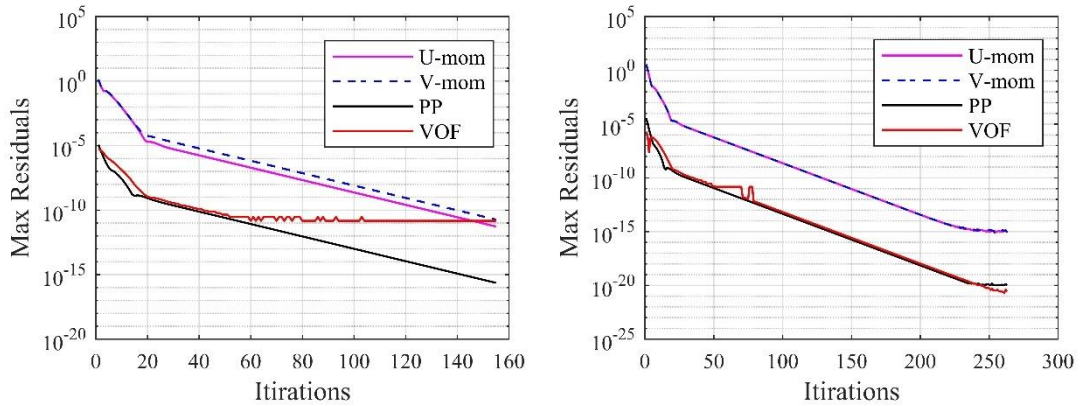


Figure 5.10: Maximum residuals obtained with redistributed CSF (left) and cell centered CSF (right) models.

### 5.2.2. Curvature Errors

Curvature was estimated directly using the volume fraction fields. The different equations to calculate curvature (Chapter IV- CSF Model- *Curvature Equations*) were compared. The original equation showed difficulty converging. Convergence was achieved with the extended equation; however, errors exist. Attempts were made to bound the curvature in order to restrict numerically miscalculated curvature in non-interfacial regions, but errors persisted.

Figure 5.11 illustrates the significant increase in maximum velocity magnitude, from  $O(10^{-15})$  to  $O(10^{-5})$ , when calculating the curvature based on the volume fraction field. This spurious current generation is attributed to the erroneous radius of curvature calculation depicted in Figure 5.12, which increases for fine meshes which is consistent with the findings of Harvie et al. [62]. Consequently, these inaccuracies lead to an inadequate representation of the pressure jump, as shown in Figure 5.13. Therefore, results highlight again the inability of adequately calculating curvature directly using VOF function.

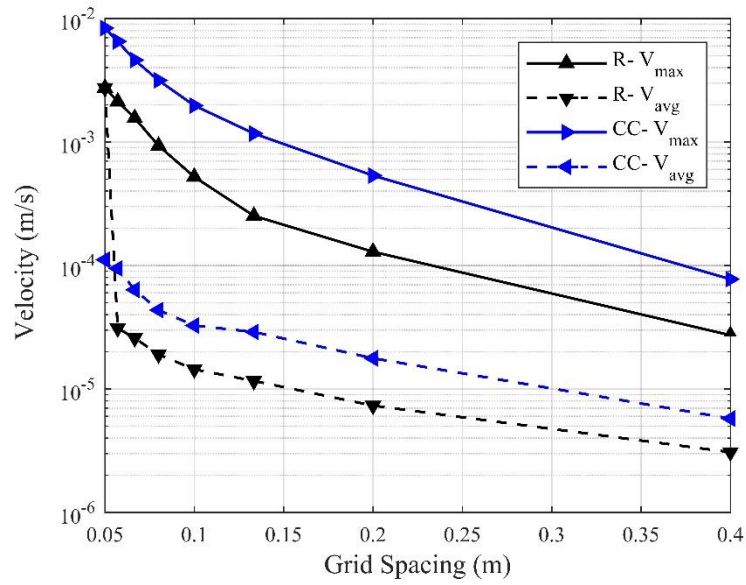


Figure 5.11: Velocity magnitude as a function of grid spacing when curvature is numerically estimated directly from VOF field.

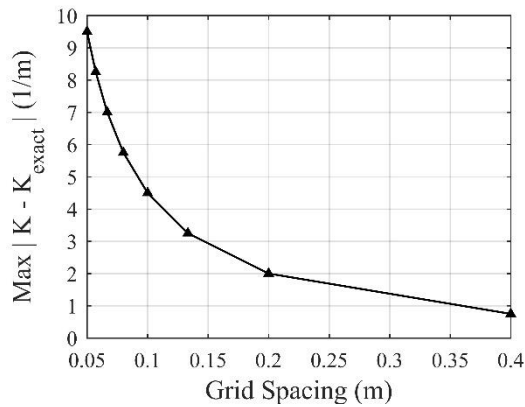


Figure 5.12: Errors in radius of curvature estimation as a function of grid spacing.

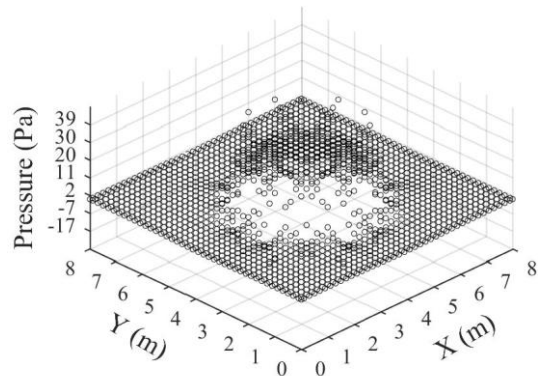


Figure 5.13: Errors in pressure jump as a function of grid spacing.

### 5.2.3. Laplacian Filter

The Laplacian filter was applied to smooth the volume fraction field prior to calculating the curvature, which showed improvement as compared to the non-smoothed curvature. Figures 5.14 and 5.15 demonstrate the effect of applying the Laplacian filter up to ten times on curvature calculation and average velocity magnitude, respectively. Reduction in spurious current generation and decreased errors in radius of curvature

estimation is observed up to eight times smoothing, beyond which minimal change is observed. Similarly, the same trend of decreasing errors can be seen with different grid spacing, as depicted in Figures 5.16 and 5.4. However, it is worth mentioning that the results converge for course meshes across the different tested number of smoothing.

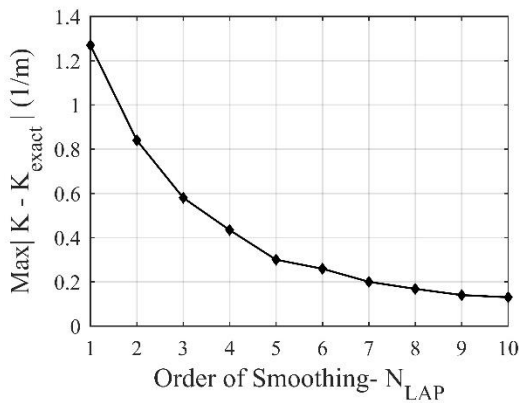


Figure 5.14: Effect of Laplacian smoothing on curvature estimation

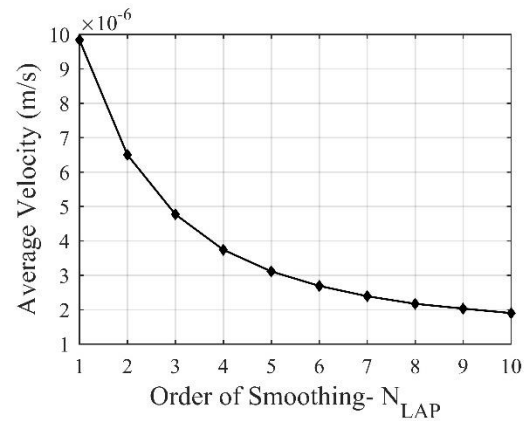


Figure 5.15: Effect of Laplacian smoothing on spurious current generation

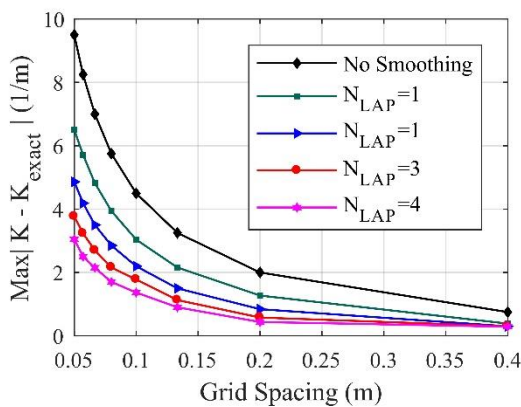


Figure 5.16: Errors in Curvature estimation as a function of grid spacing using Laplacian Filter.

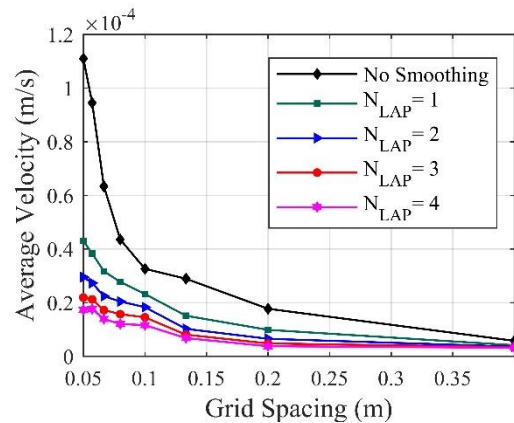


Figure 5.4: Spurious current generation as a function of grid spacing using Laplacian Filter.

#### 5.2.4. Convolution

Smoothing the volume fraction through convolution with the different kernels mentioned in Chapter IV- Convolution- *Choice of Kernel* was tested. The results of the best kernels are plotted below, which exhibit comparable errors and behavior to the work

of Francois et al. [30]. Figure 5.18 shows the maximum error in curvature as a function of support length and grid spacing, respectively. As the literature suggests, the errors in curvature estimation decrease as the size of stencil used to smooth the volume fraction increases, up to a smoothing length of nine elements, which Williams and Cummins [46, 34] identified as the required support length. Moreover, the errors resulting from different kernels converge at this support length, and further smoothing results in minimal difference in errors.

Using a stencil of nine elements, the errors in curvature estimation, velocity magnitude, and total pressure are plotted below as a function of grid spacing (figures 5.19, 5.20, and 5.21). Although the accuracy of the solution is enhanced with the convoluted color field, it is apparent that the error diverges upon mesh refinement, as reported by Evrard et al. [63]. The maximum errors in curvature estimation, across different meshes, agree with the findings of Francois et al. [30].

In addition, the maximum velocity magnitude attained after one timestep is  $5.34 \times 10^{-4}$  using K8 kernel and a support length of  $d = 9\Delta x$ . This compares to  $8.55 \times 10^{-2}$  and  $1.18 \times 10^{-3}$  reported by Francois et al. [30] and Denner [8], respectively. Spurious currents are slightly decreased with the K6 kernel,  $5.30 \times 10^{-4}$ , which also compares to  $4.87 \times 10^{-3}$  and  $1.08 \times 10^{-3}$  in the work of Francois [30] and Denner [8], respectively.

On the other hand, the pressure jump across the interface shows no clear trend for the different kernels tested. However, high pressure peaks are observed inside the drop [30] which limits the errors in maximum pressure jump to  $O(10^{-1})$  as compared to the total pressure jump of  $O(10^{-3})$ .

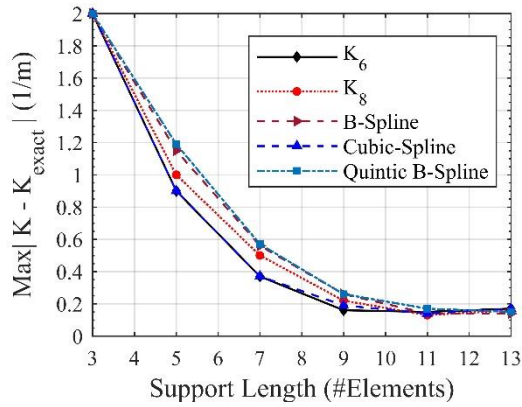


Figure 5.18: Errors in curvature estimation as a function of support

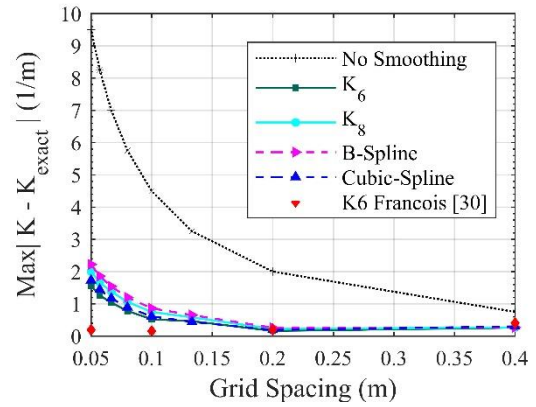


Figure 5.19: Errors in curvature estimation as a function of grid

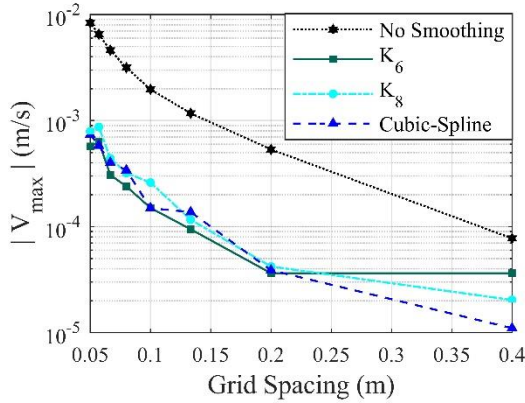


Figure 5.20: Spurious current generation as a function of grid spacing.

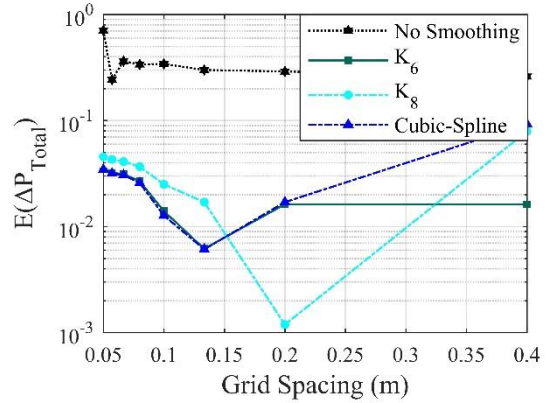


Figure 5.21: Errors in total pressure jump as a function of grid spacing.

### 5.2.5. Height Functions

The volume fraction field has been used to build height functions which in turn were used to estimate the curvature. This improved curvature estimates as compared to the original case, however, errors persisted. Height functions have been reported to produce inconsistent estimates where the interface is not accurately captured [8]. Smoothing the volume fraction field using the Laplacian filter prior to constructing height function has shown significant improvement, regardless of the stencil size used, as depicted in figure 5.22.



Using stencils of 5 and 9 elements the errors in curvature estimation, maximum velocity magnitude, and total pressure jump are plotted as a function of grid spacing in Figures 5.23, 5.24, and 5.25, respectively. Both stencils yielded similar results, surpassing the performance of the original formulation. The maximum velocity attained at 40 elements is  $1.3 \times 10^{-4}$  using 5 element stencils which compares to  $5.1 \times 10^{-3}$  and  $3.38 \times 10^{-3}$  reported by Francois et al. [30] and Liovic et al. [64]. These discrepancies are primarily attributed to curvature errors, which are particularly pronounced along non-mesh-aligned (diagonal) directions [30].

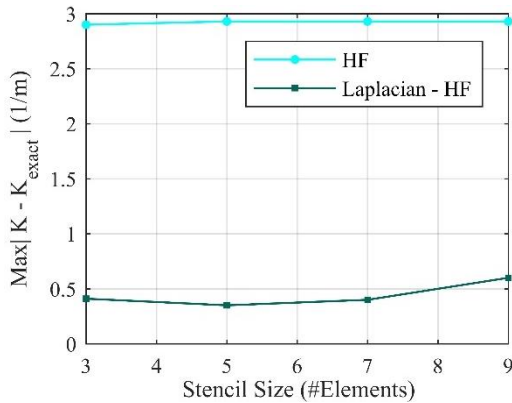


Figure 5.22: Errors in curvature estimation as a function of stencil size using HF.

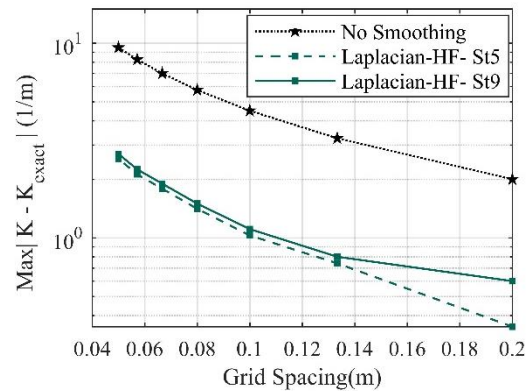


Figure 5.23: Errors in curvature as a function of grid spacing using HF.

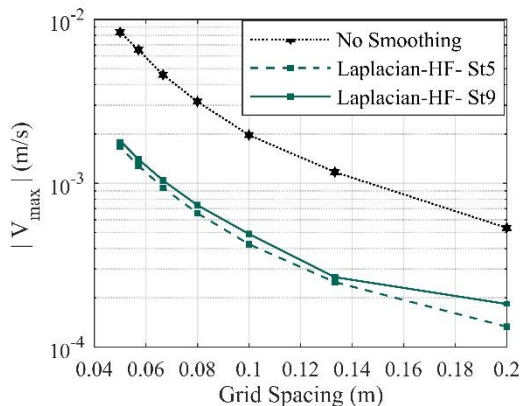


Figure 5.24: Spurious current generation as a function of grid spacing using HF.

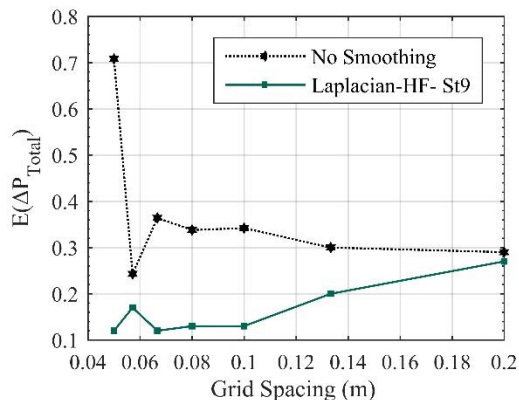


Figure 5.25: Errors in total pressure jump as a function of grid spacing using HF.

### 5.2.6. Comparing Different Techniques

The different techniques used to estimate curvature, discussed in the previous section, are grouped below in figures 5.26, 5.27, and 5.28. The results show that regardless of smoothing techniques used the results are improved as compared to the non-smoothed case which is expected because lower errors in second order derivatives are associated with smoother VOF fields. The different techniques produced similar trends for the tested parameters. The convolution technique using a K6 kernel showed the best overall results. However, while certain enhanced methods can mitigate the curvature errors when, it is not possible to extirpate it.

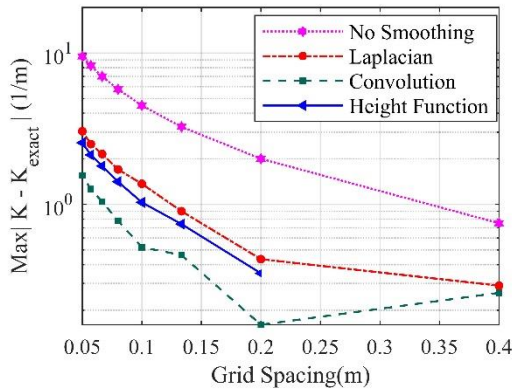


Figure 5.26: Curvature errors as a function of grid spacing.

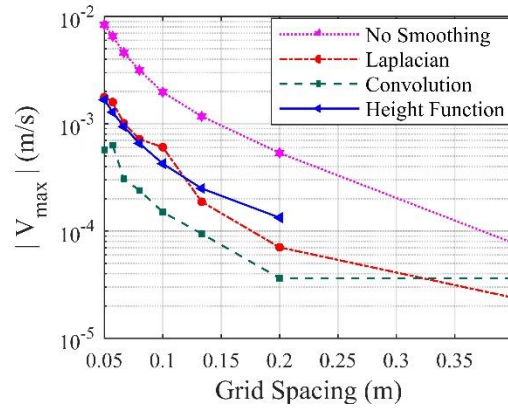


Figure 5.27: Spurious current generation as a function of grid

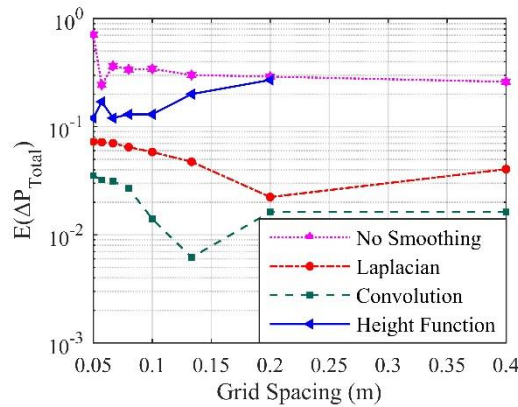


Figure 5.28: Errors in total pressure jump as a function of grid spacing.



### 5.2.7. Time Evolution

In this section the effect of time step dependency is investigated. The choice of time step to advect the color function is crucial to the accuracy and convergence of the simulation. The capillary timestep constraint, given in the equation below, sets an upper limit on the timestep size in simulations involving capillary or interfacial phenomena [8, 30, 33]. This ensures that the dynamics of these processes, occurring on very short timescales, are accurately captured.

$$\Delta t \leq \sqrt{\frac{(\rho_h + \rho_l)(\Delta x)^3}{4\pi\sigma}} \quad (5.5)$$

Baring this in mind, the same static drop as the previous section is now considered with a grid of  $R/h = 10$  and density ratio of 10. The convolution method is chosen to smooth the volume fraction with K8 kernel and support length of  $d = 9\Delta x$ . Results of the maximum velocity magnitude are obtained at  $t = 10^{-3}s$  using different time steps and tabulated below (table 3). The maximum velocity attained is of the same order  $O(10^{-3})$  as Francois et al. [30] regardless of the timestep chosen.

Table 3: Spurious current generation at  $t = 10^{-3}s$

$\Delta t$	# $\Delta t$	<b>K8</b>	<b>K8 Francois [30]</b>
1.00E-03	1	6.12E-03	1.94E-03
1.00E-04	1.00E+01	5.98E-03	1.90E-03
1.00E-05	1.00E+02	5.93E-03	1.84E-03
1.00E-06	1.00E+03	6.59E-03	1.81E-03

### 5.3. Static Bubble at Equilibrium

A static bubble of air in a water cavity is considered. The dimensions are now scaled down to a 0.02m square domain with a bubble of radius 0.005m at its center. The surface tension coefficient is set to  $7.2 \times 10^{-2}$  N/m. In this case attention is shifted to the

time evolution of the solution. The total kinetic energy Eq. (5.6), which is a quantitative measure of total kinetic energy, was evaluated at each time step.

$$TKE = \sum_1^N \frac{1}{2} \rho_c \|V\|^2 V_c \quad (5.6)$$

The case corresponds to the work of Mencinger et al. [45], where convolution with K8 kernel and the PROST techniques were employed and compared. Similarly, the K8 kernel is used, the timestep is specified at  $10^{-4}$ s, and the bubble was simulated for 4 seconds. Results of the maximum velocity magnitude as well as total kinetic energy are plotted in figures 5.29 and 5.30, respectively, and compared to the work of [45].

The results agree with the literature, with equilibrium being achieved after 0.5s, after which no change is observed. However, the present solver was able to achieve lower velocity magnitude and, consequently, lower total kinetic energy.

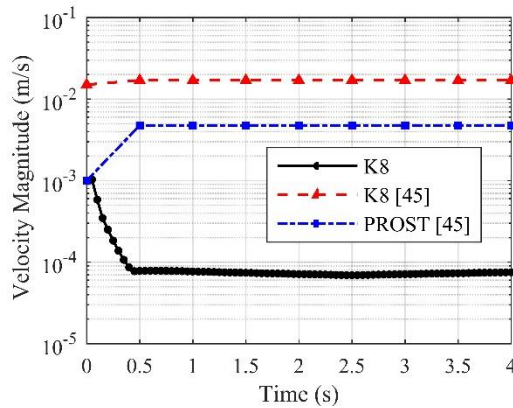


Figure 5.29: Maximum velocity magnitude as a function of time.

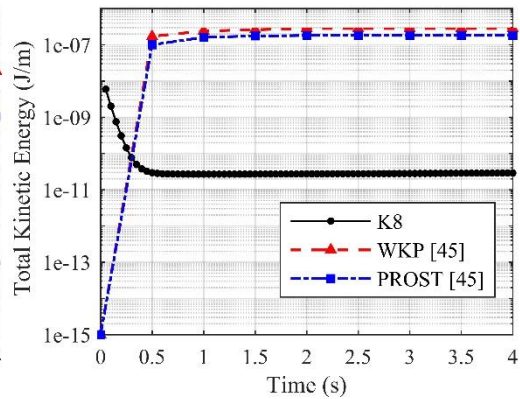


Figure 5.30: Total kinetic energy as a function of time.

#### 5.4. Breaking Dam of Water

The water dam break is one of the classical benchmark cases used to validate free surface flow modeling techniques [42]. In the literature, both experimental [65, 66] and

numerical [18, 42, 66] results can be found for this case. The problem setup is relatively straightforward: a square tank with sides measuring  $4a$  (where  $a = 0.146\text{m}$ ) is considered. Initially, a water column of size  $2a \times a$  is present in the left corner of the tank, while the rest of the domain is filled with air. The water and air characteristics are specified at room temperature and normal atmospheric pressure. The challenge in this problem arises with complex geometry carried by the advancing and returning waves [10].

As the vertical wall is removed or the simulation is initiated, gravitational forces begin to pull water towards areas of lower potential energy. As a result, the water column breaks, initiating the process of gaining momentum. This leads to the formation of advancing waves that propagate through the tank until it collides with the right side of the tank. At this point, the water climbs up the right side before falling back down, regenerating momentum and giving rise to a reverse wave. The flow dynamics involved in this scenario are primarily governed by inertia forces, as noted by Ubbink [42]. In addition, due to the large-scale nature of the problem, surface tension forces can be neglected [42]. However, as the water falls back, forming the reverse waves, air is trapped, resulting in the formation of bubbles. It is at this point that the significance of surface tension forces comes into play.

To investigate the influence of surface tension on the dam break scenario, simulations were conducted using two approaches: one neglecting surface tension and the other incorporating it through the use of CSF method, as applied in the previous test cases. In figure 5.31, the time evolution of the volume fraction field is presented, with the left column representing the case without surface tension (indicated by dark blue) and the right column representing the case with surface tension accounted for (depicted by lighter blue). Both cases exhibit similar trends and behaviors over the one-second time frame, as

expected. However, it is notable that when surface tension is considered, the formation of the reverse wave occurs more rapidly (at 0.6s instead of 0.7), aligning closer with the results reported in [42]. Furthermore, bubble formation was not observed in either case which is probably due to the large grid size used (80x80) which exceeds the expected bubble size.

To quantify the results, the height of the residual column, which refers to the bulk of the fluid excluding the waves, as well as the length to the leading edge of the wave were recorded at each time step. The results are presented in non-dimensional units given in the following relations:

$$Z = z/a \quad (5.7)$$

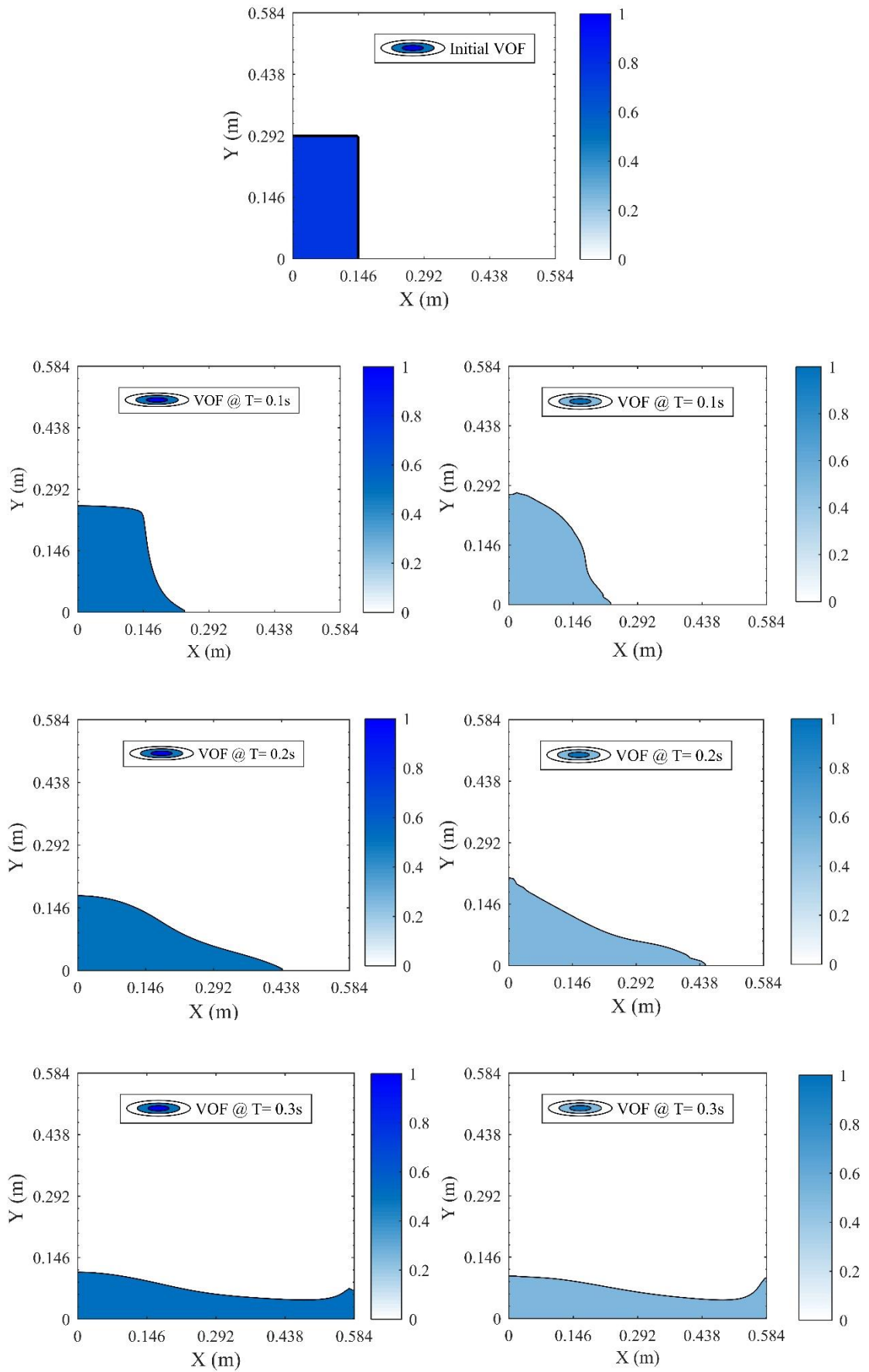
$$T = n t \sqrt{g/a} \quad (5.8)$$

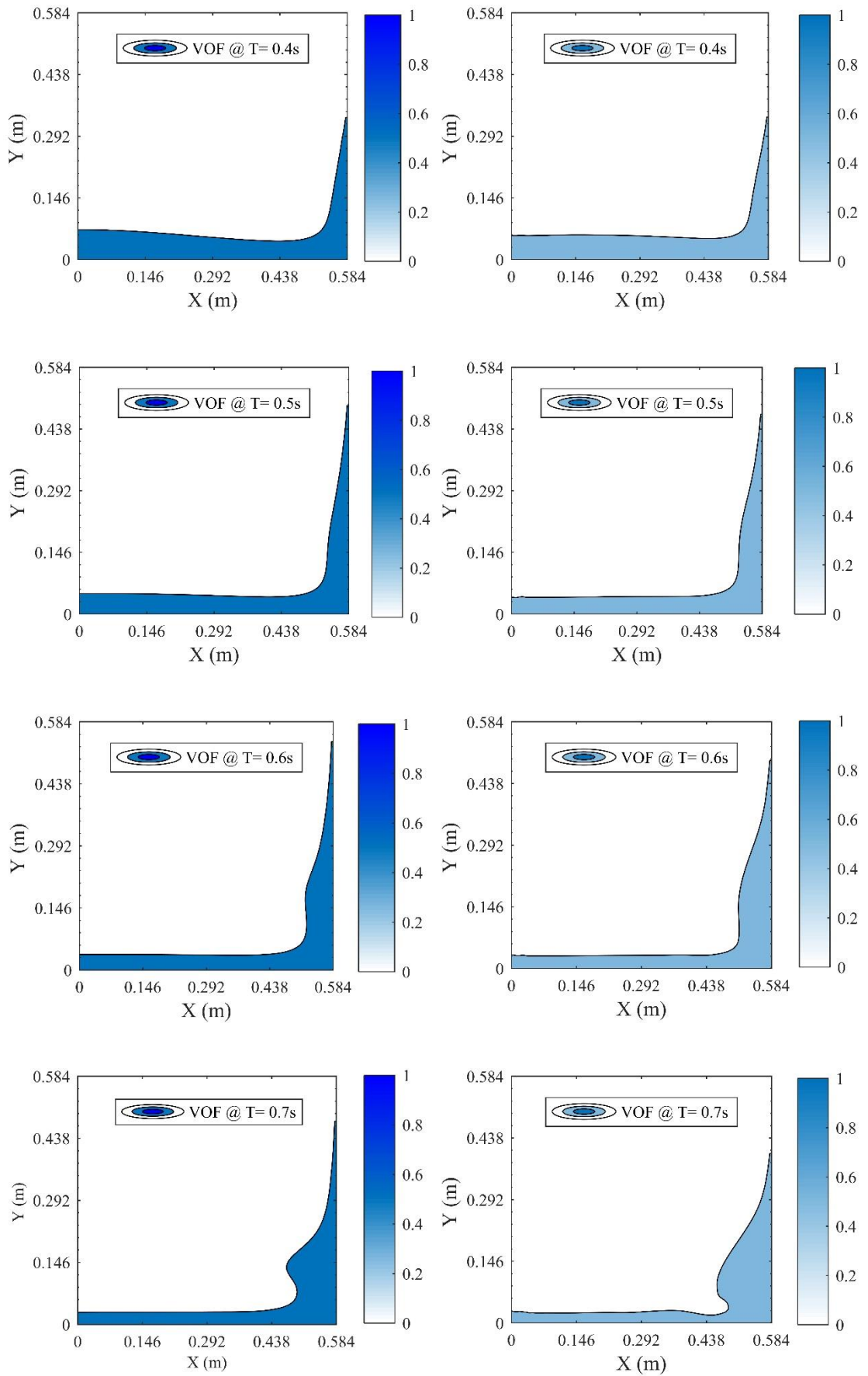
$$H = \eta/an^2 \quad (5.9)$$

$$\tau = t \sqrt{g/a} \quad (5.10)$$

Where  $a$  is the original water column base length,  $an^2$  is the original water column height,  $\eta$  is the residual column height,  $z$  is the distance to the surge front,  $g$  is the gravitational acceleration ( $9.8\text{m/s}^2$ ), and  $t$  is time (seconds).

The non-dimensional distance to the leading edge is plotted in figure 5.32 and compared to the experimental results of Martin and Moyce [65] as well as the numerical results of Hirt and Nichols [18]. The obtained results show good agreement with both experimental and simulated data. Water reached 75% of the horizontal plane after 0.2s which also agrees with the numerical results reported by Ubbink [42]. Similarly, figure 5.33 depicts the residual column height as a function of non-dimensional time, exhibiting a strong agreement with the experimental findings.





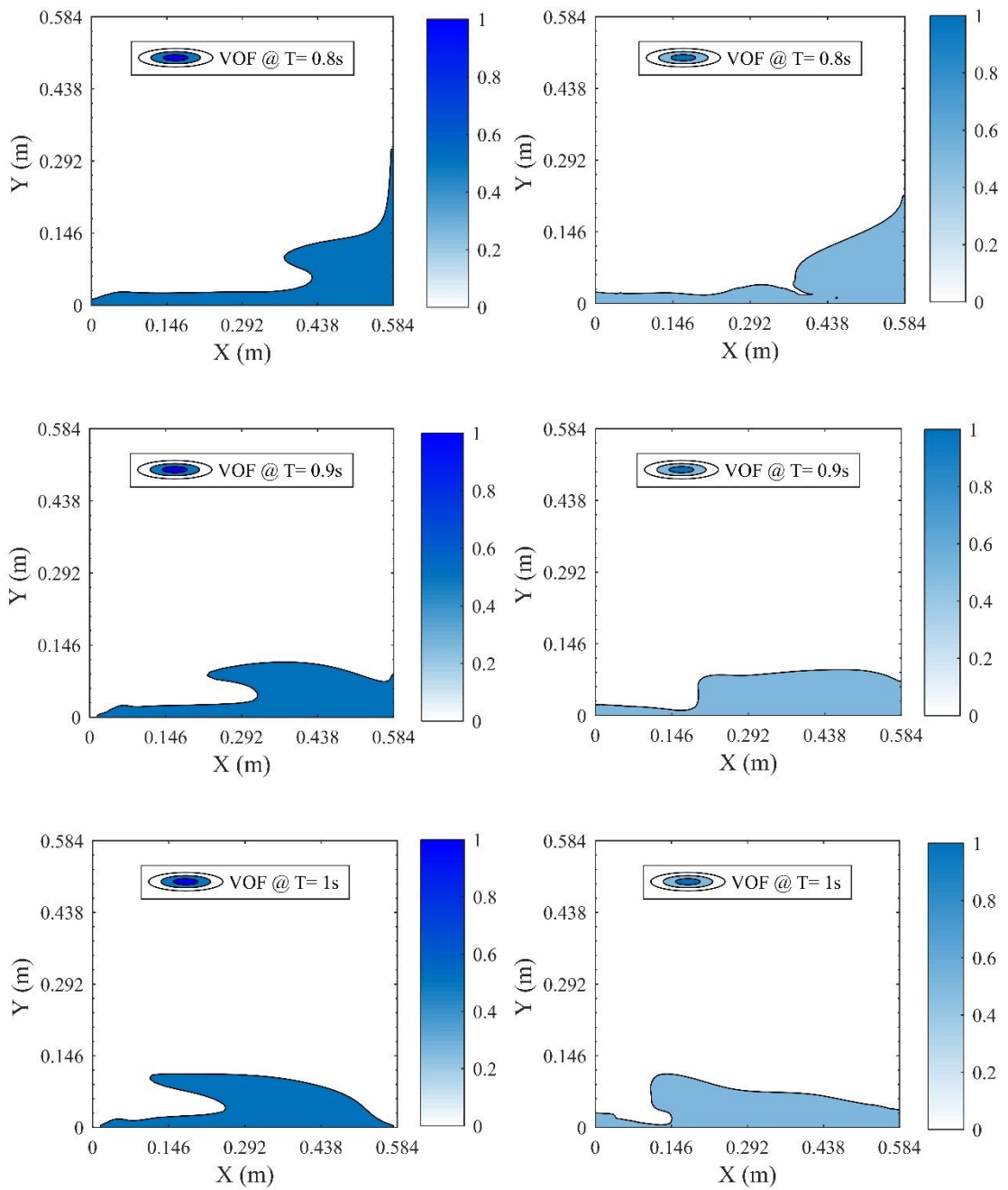


Figure 5.31: Time evolution of volume fraction field for both cases excluding surface tension (left column- dark blue) and including surface tension (right column- lighter blue).

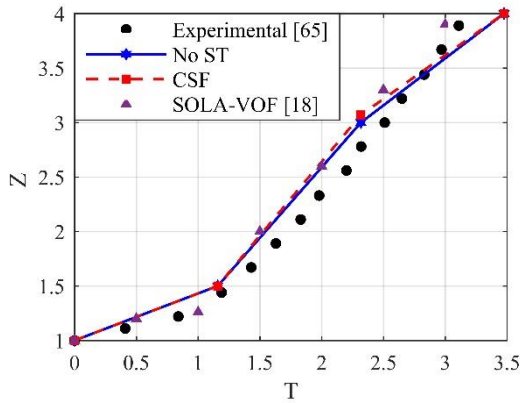


Figure 5.32: Non-dimensional distance to leading edge versus non-dimensional time.

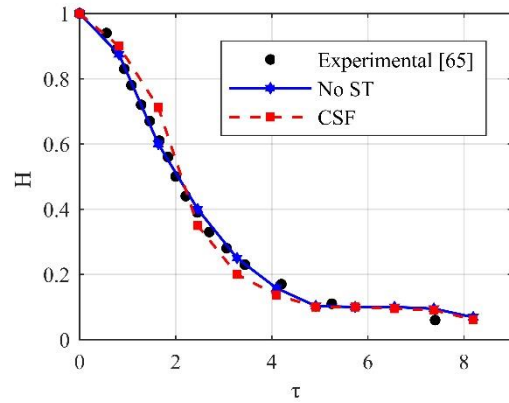


Figure 5.33: Non-dimensional residual column height versus non-dimensional time.

### 5.5. Forced Sloshing in Water Tank

Another problem involving water impact on flat surfaces and waves, though on a smaller scale, is the forced water sloshing case. It was originally introduced by Reed et al., 1998 [67], and it is usually used to tune parameters involved in liquid dampers, refer to figure 5.34.

Initially water rests at the bottom of the  $0.6 \times 0.5\text{m}^2$  tank. The tank then oscillates horizontally with a sinusoidal acceleration given in the following equation:

$$a = A\omega^2 \sin \omega t \tag{5.11}$$

Where  $A = 0.02\text{m}$  &  $\omega = 2.877\text{rad/s}$ .

From a numerical perspective, and for practical reasons, this can be implemented by introducing gravitational forces in the x-direction that obeys the same equation rather than moving the boundary. Gravitational acceleration in y-direction remains constant at  $9.8\text{m/s}^2$ . The height of water at the left-hand side of the tank is recorded throughout the simulation and plotted in figure 5.35 along with experimental data of Reed et al. [67]. Results show a comparable behavior to experimental ones and no observable difference



between the case with and without surface tension, which can also be confirmed by comparing the volume fraction evolution for the first six seconds in figure 5.36.

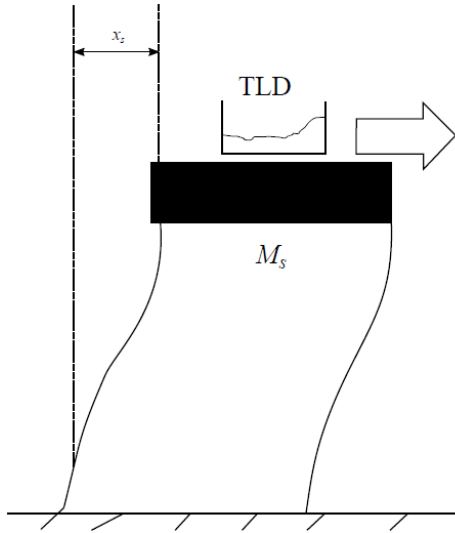


Figure 5.34: Schematic of water sloshing problem

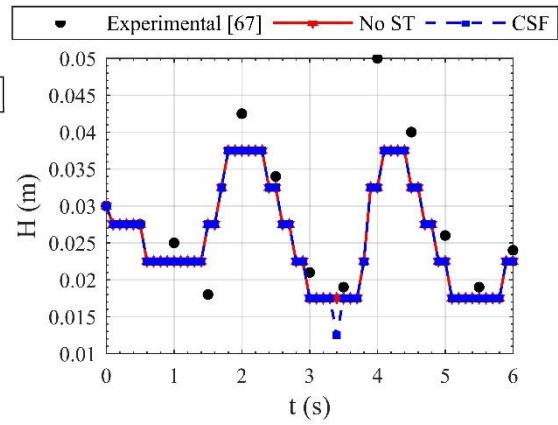
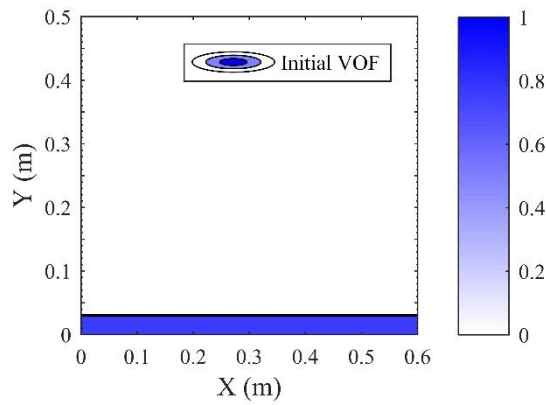
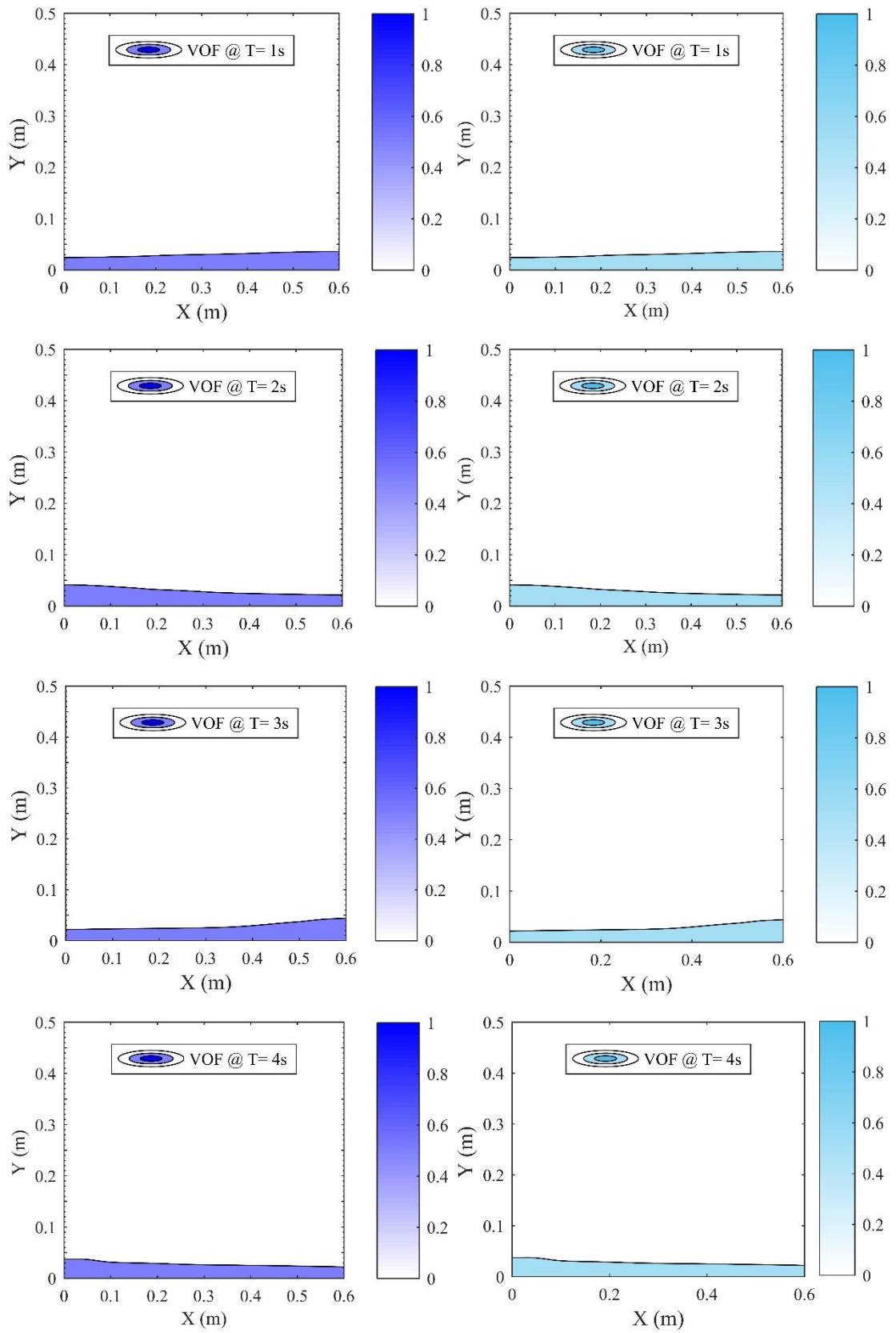


Figure 5.35: Water height at the left-hand-side of the tank





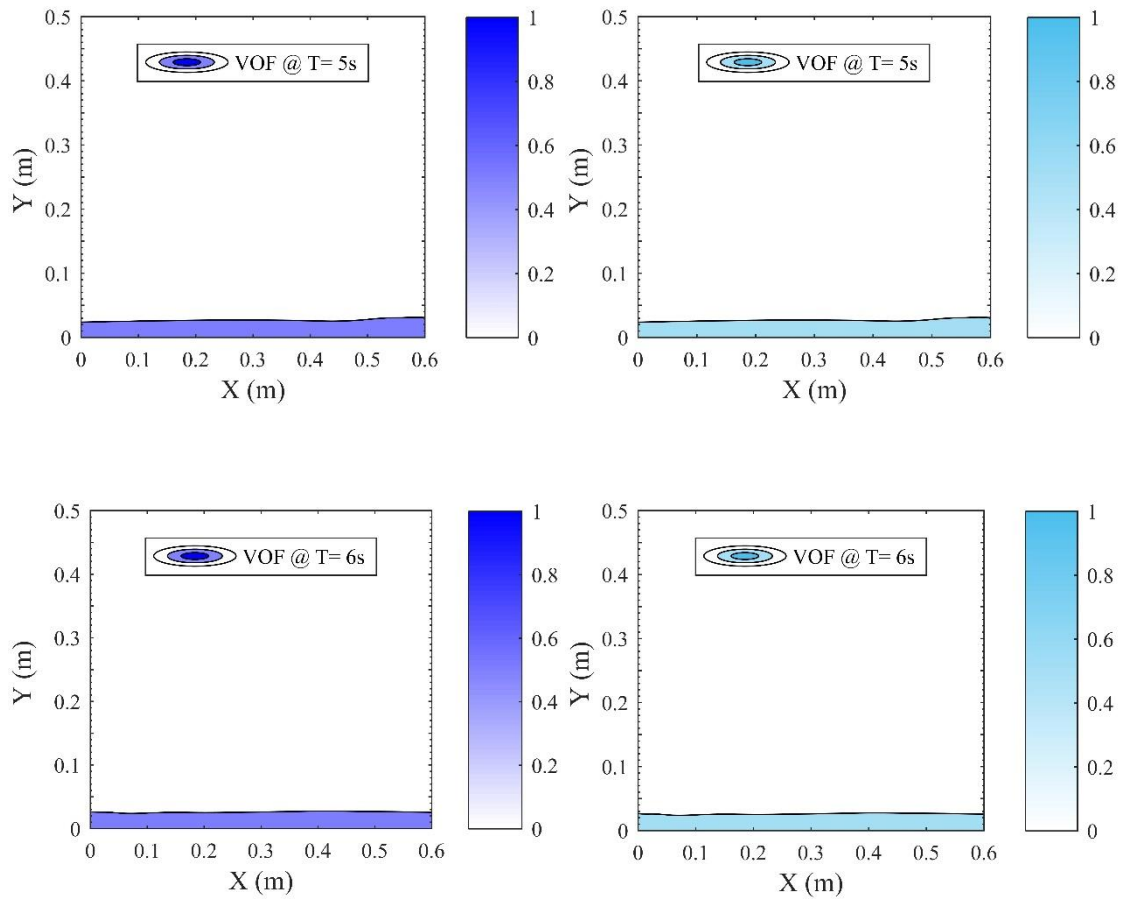


Figure 5.36: Time evolution of volume fraction field for both cases excluding surface tension (left column- dark blue) and including surface tension (right column- lighter blue).

## CHAPTER VI

### CONCLUSION

A balanced-force algorithm has been developed and tested. Different treatment of the body force term and momentum interpolation were considered. The proposed cell-centered formulation showed superior results to other formulations, with pressure jump exactly represented and spurious currents of the order of round-off.

Furthermore, the thesis explores the estimation of curvature, which cannot be directly obtained from the abruptly varying volume fraction field. Three well-known curvature estimation techniques within the Volume of Fluid (VOF) framework—namely, Laplacian Filter, Convolution Method, and Height Functions—were implemented and refined to improve the accuracy of the solution. Through extensive validation using benchmark cases, the algorithm demonstrated its effectiveness in predicting free surface flow and capturing the interface. The refined parameters of the curvature estimation techniques contributed to enhancing the overall accuracy of the results.

The contributions of this thesis offer valuable insights into simulating free surface flow with surface tension on general curvilinear coordinate systems. The balanced-force algorithm presented here provides a robust framework for future studies in this field. Further research can focus on refining the curvature estimation techniques and exploring additional methods to optimize the simulation process.

## BIBLIOGRAPHY

- [1] Hirsch, C. (2007). *Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics*. Elsevier.
- [2] Moukalled, F., Mangani, L., & Darwish, M. (2016). *The finite volume method in computational fluid dynamics* (Vol. 113, pp. 10-1007). Berlin, Germany: Springer.
- [3] LeVeque, R. J. (2002). *Finite volume methods for hyperbolic problems* (Vol. 31). Cambridge university press.
- [4] C. T. Crowe, *Multiphase flow handbook*, vol. 59. CRC press, 2005.
- [5] Katopodes, N. D. (2018). *Free-surface flow: Environmental fluid mechanics*. Butterworth-Heinemann.
- [6] Darwish, M., & Moukalled, F. (2006). Convective schemes for capturing interfaces of free-surface flows on unstructured grids. *Numerical heat transfer, part B: Fundamentals*, 49(1), 19-42.
- [7] Moukalled, F., & Darwish, M. (2012). Transient schemes for capturing interfaces of free-surface flows. *Numerical Heat Transfer, Part B: Fundamentals*, 61(3), 171-203.
- [8] Denner, F., & van Wachem, B. G. (2014). Fully-coupled balanced-force VOF framework for arbitrary meshes with least-squares curvature evaluation from volume fractions. *Numerical Heat Transfer, Part B: Fundamentals*, 65(3), 218-255.
- [9] Alloush, M. M. (2020). *The Development of a Fully Coupled Solver for the Solution of Free Surface Flows* (Doctoral dissertation).
- [10] Darwish, M. (2003). Development and testing of a robust free-surface finite volume method. *Faculty of Engineering and Architecture, American University of Beirut*.

- [11] A. Faghri and Y. Zhang, *Transport phenomena in multiphase systems*. Elsevier, 2006.
- [12] Menon, S., Mooney, K. G., Stapf, K. G., & Schmidt, D. P. (2015). Parallel adaptive simplicial re-meshing for deforming domain CFD computations. *Journal of Computational Physics*, 298, 62-78.
- [13] Herrmann, M. (2008). A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. *Journal of computational physics*, 227(4), 2674-2706.
- [14] Vachaparambil, K. J., & Einarsrud, K. E. (2019). Comparison of surface tension models for the volume of fluid method. *Processes*, 7(8), 542.
- [15] Ulvrová, M., Labrosse, S., Coltice, N., Råback, P., & Tackley, P. J. (2012). Numerical modelling of convection interacting with a melting and solidification front: Application to the thermal evolution of the basal magma ocean. *Physics of the Earth and Planetary Interiors*, 206, 51-66.
- [16] Noh, W. F., & Woodward, P. (1976). SLIC (simple line interface calculation). In *Proceedings of the fifth international conference on numerical methods in fluid dynamics June 28–July 2, 1976 Twente University, Enschede* (pp. 330-340). Springer, Berlin, Heidelberg.
- [17] Peskin, C. S. (1977). Numerical analysis of blood flow in the heart. *Journal of computational physics*, 25(3), 220-252.
- [18] Hirt, C. W., & Nichols, B. D. (1981). Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of computational physics*, 39(1), 201-225.
- [19] Spaans, E. (2018). Convolution in Multiphase Flow Simulations: Improving Interface Curvature Estimations.

- [20] Zhao, X., Xu, S., & Liu, J. (2017). Surface tension of liquid metal: role, mechanism and application. *Frontiers in Energy*, 11(4), 535-567.
- [21] Massarweh, O., & Abushaikha, A. S. (2020). The use of surfactants in enhanced oil recovery: a review of recent advances. *Energy Reports*, 6, 3150-3178.
- [22] Vachaparambil, K. J., & Einarsrud, K. E. (2018). Explanation of bubble nucleation mechanisms: a gradient theory approach. *Journal of The Electrochemical Society*, 165(10), E504.
- [23] Gallino, G., Gallaire, F., Lauga, E., & Michelin, S. (2018). Physics of Bubble-Propelled Microrockets. *Advanced Functional Materials*, 28(25), 1800686.
- [24] Clift, R., Grace, J. R., & Weber, M. E. (2005). Bubbles, drops, and particles.
- [25] Guan, P., Jia, L., Yin, L., & Tan, Z. (2016). Effect of bubble contact diameter on bubble departure size in flow boiling. *Experimental Heat Transfer*, 29(1), 37-52.
- [26] Nosonovsky, M., & Ramachandran, R. (2015). Geometric interpretation of surface tension equilibrium in superhydrophobic systems. *Entropy*, 17(7), 4684-4700.
- [27] Young, T. (1805). III. An essay on the cohesion of fluids. *Philosophical transactions of the royal society of London*, (95), 65-87.
- [28] Laplace, P. S. D. (1806). *Traité Mécanique Céleste*, Imprimerie de Crapelet. Paris NV-5.
- [29] Pellicer, J., Garcia-Morales, V., & Hernandez, M. J. (2000). On the demonstration of the Young-Laplace equation in introductory physics courses. *Physics Education*, 35(2), 126.
- [30] Francois, M. M., Cummins, S. J., Dendy, E. D., Kothe, D. B., Sicilian, J. M., & Williams, M. W. (2006). A balanced-force algorithm for continuous and sharp

- interfacial surface tension models within a volume tracking framework. *Journal of Computational Physics*, 213(1), 141-173.
- [31] Raeini, A. Q., Blunt, M. J., & Bijeljic, B. (2012). Modelling two-phase flow in porous media at the pore scale using the volume-of-fluid method. *Journal of Computational Physics*, 231(17), 5653-5668.
- [32] Lafaurie, B., Nardone, C., Scardovelli, R., Zaleski, S., & Zanetti, G. (1994). Modelling merging and fragmentation in multiphase flows with SURFER. *Journal of Computational Physics*, 113(1), 134-147.
- [33] Brackbill, J. U., Kothe, D. B., & Zemach, C. (1992). A continuum method for modeling surface tension. *Journal of computational physics*, 100(2), 335-354.
- [34] Cummins, S. J., Francois, M. M., & Kothe, D. B. (2005). Estimating curvature from volume fractions. *Computers & structures*, 83(6-7), 425-434.
- [35] Meier, M., Yadigaroglu, G., & Smith, B. L. (2002). A novel technique for including surface tension in PLIC-VOF methods. *European Journal of Mechanics-B/Fluids*, 21(1), 61-73.
- [36] Rudman, M. (1998). A volume-tracking method for incompressible multifluid flows with large density variations. *International Journal for numerical methods in fluids*, 28(2), 357-378.
- [37] Francois, M. M., Kothe, D. B., Denby, E. D., Sicilian, J. M., & Williams, M. W. (2003, January). Balanced force implementation of the continuum surface tension force method into a pressure correction algorithm. In *Fluids Engineering Division Summer Meeting* (Vol. 36975, pp. 537-542).



- [38] Renardy, Y., & Renardy, M. (2002). PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method. *Journal of computational physics*, 183(2), 400-421.
- [39] Mencinger, J. (2012). An alternative finite volume discretization of body force field on collocated grid. *Finite Volume Method: Powerful Means of Engineering Design*, 101-116.
- [40] Patankar, S. V. (1980). Numerical heat transfer and fluid flow, Hemisphere Publ. Corp., New York, 58.
- [41] Rhie, C. M., & Chow, W. L. (1983). Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA journal*, 21(11), 1525-1532.
- [42] Ubbink, O. (1997). Numerical prediction of two fluid systems with sharp interfaces.
- [43] Samkhaniani, N., & Ansari, M. R. (2016). Numerical simulation of bubble condensation using CF-VOF. *Progress in Nuclear Energy*, 89, 120-131.
- [44] Hoang, D. A., van Steijn, V., Portela, L. M., Kreutzer, M. T., & Kleijn, C. R. (2013). Benchmark numerical simulations of segmented two-phase flows in microchannels using the Volume of Fluid method. *Computers & Fluids*, 86, 28-36.
- [45] Mencinger, J., & Žun, I. (2007). On the finite volume discretization of discontinuous body force field on collocated grid: Application to VOF method. *Journal of Computational Physics*, 221(2), 524-538.
- [46] Williams, M. W. (2000). *Numerical methods for tracking interfaces with surface tension in 3-D mold-filling processes* (Doctoral dissertation, University of California, Davis).

- [47] Francois, M. M., & Swartz, B. K. (2010). Interface curvature via volume fractions, heights, and mean values on nonuniform rectangular grids. *Journal of Computational Physics*, 229(3), 527-540.
- [48] Lopez, J., Zanzi, C., Gomez, P., Zamora, R., Faura, F., & Hernandez, J. (2009). An improved height function technique for computing interface curvature from volume fractions. *Computer methods in applied mechanics and engineering*, 198(33-36), 2555-2564.
- [49] Huang, Y., & Xie, B. (2023). A generic balanced-force algorithm for finite volume method on polyhedral unstructured grids with non-orthogonality. *Journal of Computational Physics*, 479, 112010.
- [50] Khismatullin, D., Renardy, Y., & Renardy, M. (2006). Development and implementation of VOF-PROST for 3D viscoelastic liquid-liquid simulations. *Journal of non-newtonian fluid mechanics*, 140(1-3), 120-131.
- [51] S. Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, *Journal of Computational Physics* 228 (2009) 5838–5866.
- [52] H. Montazeri, C. Ward, A balanced-force algorithm for two-phase flows, *Journal of Computational Physics* 257 (2014) 645–669.
- [53] Youngs, D. L. (1982). Time-dependent multi-material flow with large fluid distortion. *Numerical methods for fluid dynamics*.
- [54] Cifani, P., Michalek, W. R., Priems, G. J. M., Kuerten, J. G., van der Geld, C. W. M., & Geurts, B. J. (2016). A comparison between the surface compression method and an interface reconstruction method for the VOF approach. *Computers & Fluids*, 136, 421-435.

- [55] Sussman, M.; Puckett, E.G. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*. **2000**, 162, 301–337.
- [56] Albadawi, A., Donoghue, D. B., Robinson, A. J., Murray, D. B., & Delauré, Y. M. C. (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.
- [57] Guo, Z., Fletcher, D. F., & Haynes, B. S. (2015). Implementation of a height function method to alleviate spurious currents in CFD modelling of annular flow in microchannels. *Applied Mathematical Modelling*, 39(16), 4665-4686.
- [58] S. Hardt, F. Wondra, Evaporation model for interfacial flows based on a continuum-field representation of the source terms, *Journal of Computational Physics* 227 (2008) 5871–5895.
- [59] Malik, M., Fan, E. S. C., & Bussmann, M. (2007). Adaptive VOF with curvature-based refinement. *International Journal for Numerical Methods in Fluids*, 55(7), 693-712.
- [60] Hernández, J., López, J., Gómez, P., Zanzi, C., & Faura, F. (2008). A new volume of fluid method in three dimensions—Part I: multidimensional advection method with face-matched flux polyhedra. *International Journal for Numerical Methods in Fluids*, 58(8), 897-921.
- [61] Magnini, M. (2012). CFD modeling of two-phase boiling flows in the slug flow regime with an interface capturing technique.

- [62] Harvie, D. J., Davidson, M. R., & Rudman, M. (2006). An analysis of parasitic current generation in volume of fluid simulations. *Applied mathematical modelling*, 30(10), 1056-1066.
- [63] Evrard, F., Denner, F., & van Wachem, B. (2017). Estimation of curvature from volume fractions using parabolic reconstruction on two-dimensional unstructured meshes. *Journal of Computational Physics*, 351, 271-294.
- [64] Liovic, P., Francois, M., Rudman, M., & Manasseh, R. (2010). Efficient simulation of surface tension-dominated flows through enhanced interface geometry interrogation. *Journal of Computational Physics*, 229(19), 7520-7544.
- [65] Martin, J. C., Moyce, W. J., Penney, W. G., Price, A. T., & Thornhill, C. K. (1952). Part IV. An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 244(882), 312-324.
- [66] Koshizuka, S., & Oka, Y. (1996). Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear science and engineering*, 123(3), 421-434.
- [67] Reed, D., Yu, J., Yeh, H., & Gardarsson, S. (1998). Investigation of tuned liquid dampers under large amplitude excitation. *Journal of engineering mechanics*, 124(4), 405-413.