

Aalborg Universitet



## Boiling Cooling Simulations: Accuracy and Efficiency Improvements Studies

Nujukambari, Ali Yahyaee

DOI (link to publication from Publisher):  
[10.54337/aau708594776](https://doi.org/10.54337/aau708594776)

Publication date:  
2024

Document Version  
Publisher's PDF, also known as Version of record

[Link to publication from Aalborg University](#)

Citation for published version (APA):  
Nujukambari, A. Y. (2024). *Boiling Cooling Simulations: Accuracy and Efficiency Improvements Studies*. Aalborg University Open Publishing. <https://doi.org/10.54337/aau708594776>

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

### Take down policy

If you believe that this document breaches copyright please contact us at [vbn@aub.aau.dk](mailto:vbn@aub.aau.dk) providing details, and we will remove access to the work immediately and investigate your claim.



**BOILING COOLING SIMULATIONS:  
ACCURACY AND EFFICIENCY  
IMPROVEMENTS STUDIES**

**BY  
ALI YAHYAEE NUJUKAMBARI**

PhD Thesis 2024



**AALBORG UNIVERSITY**  
DENMARK



---

---

# Boiling Cooling Simulations: Accuracy and Efficiency Improvements Studies

---

---

Ph.D. Dissertation  
Ali Yahyae Nujukambari

Submitted: January 2024

Main Supervisor: Assoc. Prof. Henrik Sørensen  
Aalborg University

Co-supervisors: Assoc. Prof. Amir Sajjad Bahman  
Aalborg University  
Assoc. Prof. Jakob Hærvig  
Aalborg University

Assessment: Professor Francesco Iannuzzo (Chairman)  
Aalborg University, Denmark  
Professor Marco Marengo  
University of Pavia, Italy  
Principal Scientist Timo Pättikangas  
Technical Research Centre of Finland Ltd, Finland

PhD Series: Faculty of Engineering and Science, Aalborg University

Department: AAU Energy

ISSN: 2446-1636  
ISBN: 978-87-94563-06-2

Published by:  
Aalborg University Open Publishing  
Kroghstræde 1-3  
DK – 9220 Aalborg Øst  
aauopen@aau.dk

© Copyright: Ali Yahyae Nujukambari

# Abstract

The drive towards miniaturization in engineering and design is a defining feature of modern technological advancements, focusing on creating smaller devices that retain or enhance functionality and performance. This compact design often leads to a significant increase in heat flux, hindering optimal system performance, especially when cooling mechanisms are inadequate. Traditional single-phase cooling systems, largely dependent on forced convection, struggle to meet the increased cooling demands of these compact, high-heat-flux designs. Their efficiency diminishes in scenarios with exceptionally high heat fluxes, such as in high-performance computing, advanced microelectronics, and power generation systems. In contrast, two-phase cooling technologies, particularly those involving boiling, show promise in managing these demanding scenarios, thanks to their exceptional heat transfer capabilities, which stem from extensive flow mixing due to bubbling and the latent heat of phase change.

Regarding boiling cooling methods, an in-depth understanding of flow and heat transfer processes is necessary during phase-change phenomena to enhance system performance and prevent unexpected off-design circumstances. This emphasizes the significance of research into thermally-driven phase-change processes.

This research focuses on two primary objectives: first, studying different methods to enhance the performance of simulation techniques for thermal phase change phenomena; and second, studying approaches for improving the performance of boiling cooling methods.

In addressing the first objective, a significant challenge in modeling flows with thermal phase change is accurately pinpointing the gas-liquid interface. The Volume of Fluid (VOF) technique remains the most widely adopted approach for characterizing interfaces in commercial and open-source CFD software. However, the use of VOF can lead to imprecise curvature computation and smeared interface prediction, resulting in non-physical velocities, particularly close to the interface. To recover accuracy in curvature computation for VOF simulation of boiling, a known method in two-phase flows that combines VOF with the level-set method for interface depiction can be

used. VOF is employed to capture the interface due to its mass-conserving nature, while the level-set method is used to compute the curvature near the interface. Another alternative way to enhance accuracy is by addressing the smeared face problem through the application of a recent geometric method called isoAdvector.

During the initial stages of this Ph.D. project, distinct methods including VOF-isoAdvector and the Coupled Level Set with VOF (CLSVOF) are incorporated into a custom OpenFOAM solver dedicated to thermal phase change. This integration facilitates a detailed comparison with the standard VOF method, traditionally used in OpenFOAM for interface capturing. The comparison, conducted across a range of thermal phase change benchmarks, aims to elucidate the strengths and limitations of each method. The research demonstrates that in most of the thermal phase change scenarios examined, the VOF-isoAdvector method delivers solutions more rapidly than the traditional MULES method, while maintaining comparable accuracy and convergence rates. Additionally, it is observed that the VOF-isoAdvector method reduces spurious currents near the interface and provides more precise curvature predictions. In contrast, the CLSVOF method, though demonstrating superior accuracy in curvature computation, entails longer computational times. The findings indicate that the CLSVOF method exhibited a marginally improved alignment with analytical models and benchmark solutions.

In the latter stages of this research, the focus transitions to enhancing boiling cooling methods, specifically through the use of nanofluids as a substitute for pure fluids. This shift in focus is motivated by the exceptional potential of nanofluids to elevate heat transfer rates and by the limited existing work in the realm of CFD studies on this subject. The existing CFD studies in this realm often neglect crucial factors like nanoparticle deposition on heated surfaces and the dynamic change in nanoparticle concentration during boiling, a factor influencing various thermophysical properties such as density, thermal conductivity, surface tension, heat capacity, and viscosity. Theoretical studies have pointed out that Brownian motion and thermophoresis are among the primary mechanisms governing this behavior, yet most CFD models either ignore these mechanisms or inadequately incorporate them.

Specifically, a comprehensive review of the literature reveals a primary gap. The larger set of CFD studies does not account for any governing equation for nanoparticle concentration, neglecting these important mechanisms of nanoparticle movement. The smaller set includes such an equation but limits its application to the vapor domain, thereby overlooking the liquid nanofluid phase, and also vapor-liquid interfacial regions.

To bridge the identified research gap, this study modifies a widely recognized governing equation, typically utilized for nanoparticle concentration in single-phase analysis, and expands it to accommodate the complexities inherent in multiphase film boiling scenarios. This expansion is done through the

adoption and integration of the well-established Continuous Species Transfer (CST) method, along with the employment of the Henry constant, within the context of nanofluid boiling research. This methodology facilitates a detailed examination of nanoparticle behaviors, especially at the vapor-liquid interface, and uncovers concentration variations in this area, which have not been extensively explored in previous studies. Further, a two-dimensional axisymmetric film boiling of water- $\text{Al}_2\text{O}_3$  on a vertical cylinder is studied to demonstrate the effectiveness of this method. Key focus areas included nanoparticle concentration and temperature in the domain. Findings revealed transient behaviors stabilizing over time, with an increase in nanoparticle concentration at the gas-liquid interface and also heated surface. Also, results showed that nanoparticle concentration markedly influenced heat transfer, with higher concentrations improving efficiency, as evidenced by increased Nusselt numbers.

While phenomena like variations in triple-line motion because of nanoparticle deposition do occur in nanofluid boiling, the case study here is deliberately chosen to be on the film boiling of water- $\text{Al}_2\text{O}_3$  nanofluid. In such a film boiling scenario, an insulating vapor film minimizes this effect by preventing direct liquid-to-surface contact.

It is essential to note that this thesis is focused on the boiling phenomenon itself, which can include both pool and flow boiling. Chapters 2 and 3 are dedicated to examining and evaluating the integration of diverse interface description methods—specifically VOF-MULES, VOF-isoAdvector, and CLSVOF—within a thermal phase change OpenFOAM solver. These chapters scrutinize well-known thermal phase change benchmark cases, some of which do not distinctly align with the conventional categories of either pool or flow boiling. Notably, the solvers developed in these chapters are capable of simulating scenarios typical of both pool and flow boiling. Chapter 4 shifts the attention to a specific case study of nanofluid film boiling. Although this chapter primarily deals with a pool boiling scenario, the model developed is adaptable and can be modified for application in flow boiling situations as well.



# Resumé

Drive mod miniaturisering i ingeniørarbejde og design er et kendetegnende træk ved moderne teknologiske fremskridt, der fokuserer på at skabe mindre enheder, der bevarer eller forbedrer funktionalitet og ydeevne. Denne kompakte design resulterer ofte i en betydelig stigning i varmeflux, hvilket hindrer systemets optimale ydeevne, især når kølingen ikke håndteres tilstrækkeligt. Traditionelle enkeltfase-kølesystemer, der overvejende er afhængige af tvungen konvektion, finder det i stigende grad udfordrende at imødekomme de forøgede kølebehov, som disse kompakte, højvarmeflux-designs stiller. Selvom de stadig er effektive i mange anvendelser, tendenser deres effektivitet til at formindskes i scenarier med ekstremt høje varmeflux, hvilket understreger behovet for mere avancerede køleløsninger. På den anden side er tofaset køleteknologi, især dem, der involverer kogning, lovende til håndtering af høje varmefluxscenarier, takket være den ekstraordinære varmeoverførselsydelse som følge af betydelig strømningblanding på grund af bobledannelse og faseændringens latente varme.

Når det gælder kogekølingsmetoder, er det nødvendigt med en dybdegående forståelse af strømninger og varmeoverførselsprocesser under faseændringsfænomener for at forbedre systemets ydeevne og forhindre uventede off-design-situationer. Dette understreger betydningen af forskning i termisk drevne faseændringsprocesser.

Denne forskning fokuserer på to primære mål: først at studere forskellige metoder til at forbedre ydeevnen af simuleringsmetoder for faseændringsfænomener; og for det andet at studere tilgange til forbedring af ydeevnen af kogekølingsmetoder.

I forbindelse med det første mål er en betydelig udfordring ved modellering af strømninger med termisk faseændring nøjagtigt at identificere gasvæske-grænsefladen. Volume of Fluid (VOF)-teknikken forbliver den mest udbredte metode til karakterisering af grænseflader i kommercielle og open-source CFD-software. Imidlertid kan brugen af VOF føre til upræcis beregning af grænsefladen, hvilket resulterer i ikke-fysiske hastigheder, især tæt på grænsefladen. For at genvinde nøjagtigheden i VOF-simuleringen anvendes en solver, der kombinerer VOF med level-set metoden til beregning. VOF

anvendes til at fange grænsefladen på grund af dens massebevarende natur, mens level-set metoden bruges til at beregne krumningen og de fysiske egenskaber nær grænsefladen. En alternativ måde at forbedre nøjagtigheden på er ved at løse problemet med grænseflade gennem anvendelsen af en relativt ny geometrisk metode kaldet isoAdvector.

I den indledende fase af dette ph.d.-projekt blev forskellige metoder, herunder VOF-isoAdvector og Coupled Level Set med VOF (CLSVOF), indarbejdet i en brugerdefineret OpenFOAM-løser til termisk faseændring. Denne integration muliggjorde en detaljeret sammenligning med den standard VOF-metode, der traditionelt anvendes i OpenFOAM til grænsefladeopsamling. Sammenligningen, der blev udført på tværs af en række benchmarks for termisk faseændring, sigtede mod at belyse styrkerne og begrænsningerne ved hver metode. Forskningen viste, at VOF-isoAdvector-metoden i de fleste af de undersøgte scenarier for termisk faseændring leverede løsninger hurtigere end den traditionelle MULES-metode, samtidig med at den opretholdt sammenlignelig nøjagtighed og konvergenshastigheder. Desuden blev det observeret, at VOF-isoAdvector-metoden reducerede spuriose strømme nær grænsefladen og gav mere præcise krøllingsforudsigelser. I modsætning her til viste CLSVOF-metoden, selvom den demonstrerede overlegen nøjagtighed i krøllingsberegning, længere beregningstider. Resultaterne viste, at CLSVOF-metoden viste en marginalt forbedret overensstemmelse med analytiske modeller og benchmark-løsninger.

I øjeblikket er forskningen fokuseret på at forbedre metoderne til køling ved kogning, specifikt gennem anvendelsen af nanofluid som erstatning for rene væsker. Denne ændring i fokus er motiveret både af nanofluiders ekstraordinære potentiale til at forhøje varmeoverførselshastigheder og stabilisere kogeprocesser, og af det begrænsede eksisterende arbejde på dette område inden for CFD-studier. De eksisterende CFD-studier på dette område overser ofte afgørende faktorer som nanopartikelaflejring på opvarmede overflader og variationer i den trefasede linjeform. En af de mest betydelige udeladelser fra nuværende CFD-modeller er den dynamiske ændring i nanopartikelkoncentration under kogning, en faktor der påvirker forskellige termofysiske egenskaber såsom densitet, termisk konduktivitet, overfladespænding, varmekapacitet og viskositet. Teoretiske studier har påpeget, at Brownsk bevægelse og termoforese er blandt de primære mekanismer, der styrer denne adfærd, men de fleste CFD-modeller ignorerer enten disse mekanismer eller inkorporerer dem utilstrækkeligt.

Specifikt afslører en omfattende gennemgang af litteraturen to primære mangler. Den større gruppe af CFD-studier tager ikke højde for nogen styrende ligning for nanopartikelkoncentration, og ignorerer således disse vigtige nanopartikelbevægelsesmekanismer. Den mindre gruppe inkluderer en sådan ligning, men begrænser dens anvendelse til dampdomænet og overser dermed den flydende nanofluid-fase.

I konteksten af filmkogning, der involverer nanofluids, danner en isolerende dampfilm mellem nanofluiden og den opvarmede overflade. Denne dampfilm er dynamisk og oplever fordampning ved dens ydre damp-væske grænseflade, hvilket fører til en stigning i nanopartikelkoncentrationen ved grænsefladen – endnu et aspekt, der altid overses i eksisterende CFD-modeller.

For at imødegå disse forskningsmangler introducerer studiet en specialiseret OpenFOAM-baseret solver, der anvender Continuous-Species-Transfer (CST) metoden inden for rammerne af Computational Multi-Fluid Dynamics (CMFD). Denne solver er unikt designet til at simulere nanopartikeloverførsler mellem damp- og væskefaser. Vigtigt er det, at den inkorporerer styringsligninger for termoforese og Brownsk bevægelse for at tage højde for ændringer i nanopartikelkoncentrationen ved damp-væske grænsefladen.

Selvom fænomener som nanopartikelaflejring og variationer i trefaselinjens bevægelse forekommer ved nanofluid kogning, er fokus her bevidst på filmkogning. I sådanne scenarier minimerer en isolerende dampfilm disse effekter ved at forhindre direkte væske-til-overflade kontakt.

Denne undersøgelse anvendte en todimensional akse-symmetrisk filmkogningsanalyse på en lodret cylinder for at demonstrere effektiviteten af denne metode. Vigtige fokusområder inkluderede nanopartikelkoncentration og temperatur i domænet. Fundene afslørede overgangsadfærd, der stabiliserede sig over tid, med en stigning i nanopartikelkoncentrationen ved gas-væske grænsefladen på grund af fordampning af dampfilmen. Desuden viste resultaterne, at nanopartikelkoncentrationen markant påvirkede varmeoverførslen, med højere koncentrationer, der forbedrede effektiviteten, som bevidnet af øgede Nusselt-tal.

Det er væsentligt at bemærke, at denne afhandling fokuserer på selve kogefænomenet, som kan inkludere både pulje- og strømningskogning. Kapitler 2 og 3 undersøger velkendte benchmark-sager, der ikke strengt falder ind under hverken pulje- eller strømningskogning kategorierne. Disse kapitler sigter mod at verificere, validere og analysere ydeevnen af forskellige grænsefladebeskrivelsesmetoder i OpenFOAM-solvere i simulering af kogning. Specifikt studerer de grænsefladebeskrivelsesmetoder, nemlig VOF-MULES, VOF-isoAdvector og CLSVOF. Kapitel 4 fokuserer derimod på en casestudie, der involverer nanofluid filmkogning. Selvom denne undersøgelse hælder mere mod strømningskogning, er den udviklede solver alsidig og kan anvendes til både pulje- og strømningskogningsscenarier.

## Resumé

# Contents

<b>Abstract</b>	<b>iii</b>
<b>Resumé</b>	<b>vii</b>
<b>Preface</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Boiling Phenomena and Stability Challenges . . . . .	3
1.1.1 Detailed Mechanism of Bubble Formation and Growth . . . . .	3
1.1.2 Boiling Types . . . . .	4
1.1.3 Limitations, and stability . . . . .	7
1.2 Gas-Liquid interface simulation . . . . .	13
1.3 Motivation . . . . .	17
1.4 Research Questions, Hypotheses, and Objectives . . . . .	18
1.5 Thesis Outline . . . . .	20
1.6 List of Publications . . . . .	21
References . . . . .	22
<b>2 isoAdvectord approach to describe the gas-liquid interface</b>	<b>27</b>
2.1 Literature review on different thermal phase change solvers in OpenFOAM . . . . .	28
2.2 Literature review on VOF-isoAdvectord method . . . . .	30
2.3 Numerical Formulation . . . . .	31
2.3.1 Governing equations . . . . .	31
2.3.2 Models for Thermal Phase change . . . . .	35
2.3.3 Discretization Methods and Solver Configuration . . . . .	36
2.3.4 Dimensionless numbers . . . . .	37
2.4 Comparison: thermalPhaseChangeFlow Solver and twoPhase- Flow Solver . . . . .	39
2.5 Benchmark cases . . . . .	41
2.5.1 Stefan problem . . . . .	42
2.5.2 Horizontal film condensation . . . . .	46

## Contents

2.5.3	Laminar film condensation on a vertical plate . . . . .	49
2.5.4	2D film boiling . . . . .	54
2.5.5	Stationary 3D Spherical Bubble Scenario . . . . .	58
2.6	Conclusion . . . . .	62
	References . . . . .	64
<b>3</b>	<b>CLSVOF method to describe the gas-liquid interface</b>	<b>67</b>
3.1	Literature review on level set and CLSVOF methods . . . . .	68
3.2	CLSVOF governing equations . . . . .	69
3.3	Benchmark Cases . . . . .	72
3.3.1	Stefan problem . . . . .	73
3.3.2	Horizontal film condensation problem . . . . .	75
3.3.3	Laminar film condensation on a vertical plate . . . . .	77
3.3.4	2D film boiling . . . . .	80
3.3.5	Stationary 3D Spherical Bubble Scenario . . . . .	82
3.4	Conclusion . . . . .	84
	References . . . . .	86
<b>4</b>	<b>Nanofluid Film Boiling</b>	<b>89</b>
4.1	Literature Review on Boiling of Nanofluids . . . . .	90
4.2	Mathematical Framework and Governing Equations . . . . .	95
4.2.1	Theoretical Framework and Methodology . . . . .	95
4.2.2	Bubble Dynamics via the Volume-Of-Fluid Method . . . . .	96
4.2.3	Modeling Nanoparticle Dynamics with CST, Brownian Motion, and Thermophoresis . . . . .	98
4.2.4	Energy Conservation Equation . . . . .	103
4.2.5	Updating Thermo-Physical Properties of Nanofluid . . . . .	104
4.2.6	Validation with Taylor 2D Film Boiling Simulations . . . . .	105
4.3	Problem Description . . . . .	109
4.4	Results and Analysis . . . . .	112
4.4.1	Nanofluid Volume Fraction . . . . .	113
4.4.2	Nanoparticles concentration . . . . .	114
4.4.3	Temperature distribution . . . . .	117
4.4.4	Nanofluid Volume Fraction and Nanoparticle Concen- tration over time . . . . .	118
4.4.5	Nanofluid thermophysical characteristics . . . . .	119
4.4.6	Nusselt number . . . . .	122
4.5	Conclusion . . . . .	123
	References . . . . .	125
<b>5</b>	<b>Conclusion</b>	<b>131</b>
5.1	Summary . . . . .	131
5.2	Future works . . . . .	133

Contents

A1

135

## Contents

# Preface

This dissertation, titled "Boiling Cooling Simulations: Accuracy and Efficiency Improvement Studies," embarks on an in-depth study, aiming to enhance the precision and efficiency of numerical CFD simulations of phase change phenomena. This research has been generously funded by the Department of Energy Technology at Aalborg University and received additional support from Otto Mønsted's Fond.

The essence of this dissertation lies in a comprehensive analysis of selected theories and methods that form the backbone of CFD simulation of boiling. Readers are assumed to have prior familiarity with CFD methods and the associated theory. The study applies these methods and undertakes a detailed investigation into their effectiveness and precision, with a broader goal to stimulate advancements in boiling cooling simulations.

I extend my profound gratitude to my supervisor, Assoc. Prof. Henrik Sørensen, for his unwavering support and guidance. His knowledge and insightful perspectives have significantly enriched this research. Equally, I wish to express my heartfelt thanks to my co-supervisors, Assoc. Prof. Jakob Hærvig and Assoc. Prof. Amir Sajjad Bahman. Their exceptional expertise, technical prowess in CFD discussions, and constant encouragement have been instrumental in deepening my understanding and navigating the intricate aspects of this project.

The intellectual exchanges and valuable feedback from my colleagues at the Department of Energy Technology, Aalborg University, have significantly enhanced my research experience. Their camaraderie has made this journey both rewarding and enjoyable.

As a special token of acknowledgment, I wish to express my deepest gratitude to my family. They have been my bedrock of support, consistently encouraging me and providing warmth and strength. Their unwavering belief in me has been a powerful source of resilience throughout this academic endeavor.

Ali Yahyae Nujukambari  
Aalborg University, December, 2023

## Preface

# Chapter 1

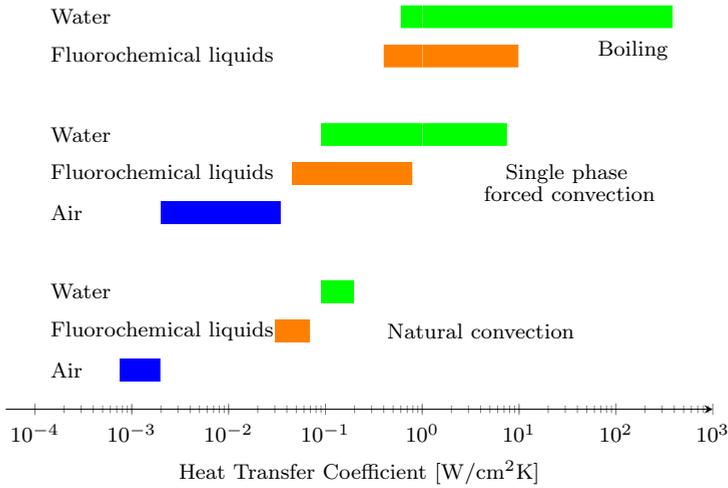
## Introduction

Miniaturization, coupled with the goal of maintaining or enhancing performance, is a key focus in today's engineering and technological sectors. This trend is most pronounced in areas such as transportation, aviation, supercomputing, and data centers, where a notable escalation in heat flux is observed in increasingly compact devices. This rise in heat generation presents substantial challenges, especially when conventional cooling mechanisms are inadequate, thereby limiting the full operational potential of these devices.

Traditional single-phase cooling strategies, while effective in various applications, are increasingly challenged by the escalating demand for efficient heat dissipation in high heat flux scenarios. The limitations of these single-phase systems become more apparent in comparison to two-phase cooling technologies, particularly those involving boiling. Boiling cooling methods offer a notably advanced solution for managing high heat flux, primarily due to their exploitation of extensive flow mixing caused by bubbling and the efficient use of the latent heat of phase change. These attributes enable boiling cooling methods to manage significantly higher levels of heat flux, making them a compelling choice for applications requiring advanced thermal management solutions.

Fig. 1.1 presents an overview of various cooling methods and their corresponding heat transfer coefficients. It categorizes the heat transfer capabilities of water, fluorocarbon liquids, and air under natural convection, single-phase forced convection, and boiling conditions. This illustration highlights the broad range of heat transfer coefficients achievable through different cooling strategies, with boiling conditions offering significantly higher coefficients indicative of more efficient heat removal capacity. This method's superior performance is crucial in applications where robust heat dissipation is paramount. Attaining an in-depth and detailed understanding of the flow and heat transfer mechanisms during phase-change events is vital for

## Chapter 1. Introduction



**Fig. 1.1:** Overview of diverse cooling approaches and associated heat transfer coefficient ranges [1]

enhancing system performance and preventing unforeseen off-design situations. This recognition of their crucial role justifies their relevance and motivates further studies to refine their precision and effectiveness, forming the cornerstone of this dissertation.

Boiling heat transfer research typically involves both experimental testing and numerical modeling. While experimental studies are crucial for obtaining precise measurements in specific areas of study, they come with notable challenges, including high costs and complexity. These studies often necessitate the use of numerous sensors for data collection, which can be expensive and generally provide only a limited amount of data. Additionally, the high pressures encountered in boiling heat transfer experiments can limit the scope of research. This is particularly true in micro-scale applications that require high heat fluxes in small spaces, where creating appropriate experimental setups can be highly challenging.

In contrast, numerical simulations in boiling heat transfer studies can present significant advantages, especially as they evolve and become more refined. These simulations can effectively overcome many of the constraints associated with experimental methods, enabling the acquisition of detailed and extensive data across a wide range of conditions and applications. Numerical models can facilitate in-depth analysis and visualization of complex boiling phenomena, which are often difficult or even impossible to capture through experimental means. Accordingly, the efforts presented in this Ph.D. dissertation will primarily focus on advancing and refining numerical methods for the analysis of boiling processes.

## 1.1 Boiling Phenomena And Stability Challenges

Boiling is a critical heat transfer process widely utilized in numerous industrial and engineering settings. Its ability to handle high heat fluxes effectively makes it a key component in thermal management across various systems, ranging from power generation to the cooling of electronics. This section of the thesis examines the complex mechanisms that underpin the boiling process. It begins with an exploration of the basic dynamics involved in the formation and growth of bubbles. The discussion then extends to different types of boiling, each with its flow patterns and thermal characteristics. Additionally, the section addresses the significant stability challenges that are often encountered in boiling systems and explores potential strategies to mitigate these issues.

### 1.1.1 Detailed Mechanism Of Bubble Formation And Growth

The boiling process encompasses the nucleation and expansion of vapor bubbles on a heated surface, which starts when the surface temperature exceeds the liquid's saturation temperature, as depicted in Fig. 1.2. This section elucidates the sequential stages of this complex heat transfer mechanism.

- **Nucleation at Active Sites:** Bubble nucleation typically begins at active sites on the heating surface. These sites are microscopic imperfections such as scratches, cavities, or material defects that facilitate the formation of vapor bubbles by providing energy localization.
- **Microlayer Evaporation:** A microlayer of liquid forms between the heated surface and the base of the nucleating bubble. Intense heat transfer from the surface induces rapid evaporation of this microlayer, contributing to the initial growth of the vapor bubble.
- **Triple Line Evaporation:** Alongside the microlayer evaporation, the triple line – the juncture where solid, liquid, and vapor meet – also experiences evaporation. This contributes to the vapor accumulation and further growth of the bubble.
- **Surface Evaporation:** Surrounding the developing bubble, surface evaporation contributes to the ongoing growth, as the liquid nanofluid transitions to vapor, adding to the bubble's volume.
- **Forces Governing Bubble Dynamics:** The growth of the bubble is governed by the interplay of buoyancy and surface tension forces. Buoyancy drives the bubble upward, while surface tension aims to minimize the bubble's surface area.

- **Detachment and Ascension:** The bubble detaches from the surface when buoyancy surpasses the restraining surface tension forces, allowing the bubble to ascend to the liquid's surface and release its vapor content.

This description captures the critical role of active sites and various evaporative phenomena in bubble formation and growth, which collectively dictate the boiling heat transfer efficiency.

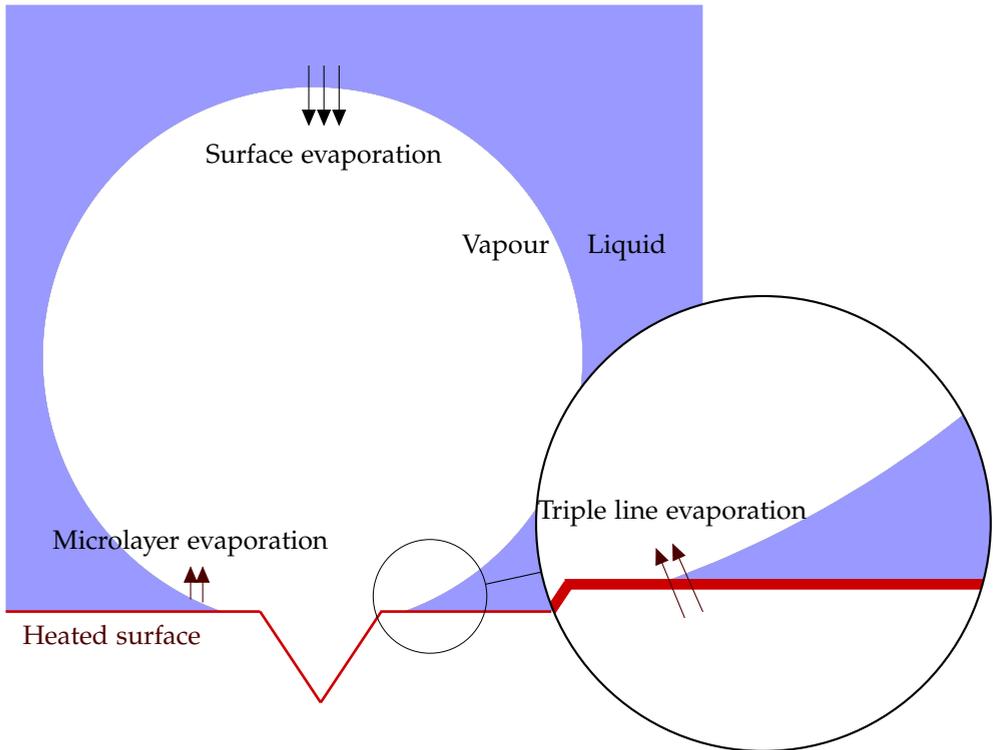


Fig. 1.2: Bubble nucleation and development at a hot surface.

## 1.1.2 Boiling Types

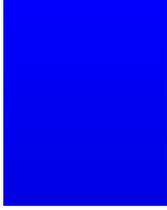
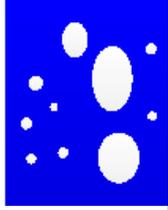
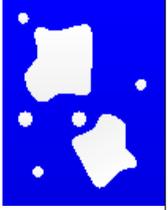
Boiling can be broadly categorized into two types: pool boiling and flow boiling. Each type exhibits unique flow patterns and regimes that significantly influence heat transfer rates.

### Pool Boiling

Pool boiling is characterized by the formation of vapor bubbles on heated walls and the existence of quiescent liquid. The flow patterns in pool boiling

## 1.1. Boiling Phenomena and Stability Challenges

evolve through several regimes, as illustrated in Fig. 1.3, and are detailed below:

Single Phase Convection	Nucleate Boiling		Transition Boiling	Film Boiling
Pure Liquid	Isolated Bubbles	Jets and Columns	Distributed Slug Flow	Annular Flow
				

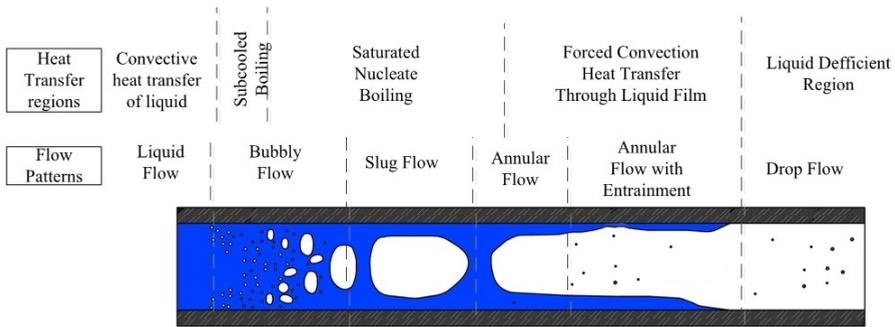
**Fig. 1.3:** Regimes of Pool Boiling: A visual comparison of the different flow patterns observed in the pool boiling on heated walls of a vertical channel, from single-phase convection to film boiling.

- Nucleate Boiling:** This regime starts with the formation of isolated bubbles at nucleation sites on the heated surface. These bubbles are created when the surface temperature exceeds the liquid's saturation temperature. As the heat flux increases, the frequency and size of the bubbles increase, leading to the formation of jets and columns of vapor. In this regime, the heat transfer coefficient is generally high due to efficient liquid-vapor exchange. This makes nucleate boiling highly effective for thermal management applications [2].
- Transition Boiling:** Transition boiling occurs between the nucleate and film boiling regimes. In this stage, the system experiences a mix of characteristics from both regimes. The surface is partly covered with vapor patches, leading to a disturbed flow pattern akin to slug flow caused by buoyancy effects. The heat transfer rates in this regime become unstable and erratic, and the heat transfer coefficient can vary significantly. This instability is due to the irregular formation and collapse of vapor patches on the surface [2].
- Film Boiling:** In film boiling, a continuous and stable vapor layer forms over the heated surface, effectively insulating it from the liquid. This regime typically occurs at higher surface temperatures. The vapor film behaves like an annular flow, where the inner layer is a vapor film and the outer layer is the bulk liquid. Due to the insulating nature of the vapor film, the heat transfer rate is significantly reduced compared to

nucleate boiling. Film boiling is less efficient for heat transfer, but understanding its dynamics is crucial for safety in high-temperature applications [2].

## Flow Boiling

Flow boiling is characterized by the forced convective flow of a liquid over a heated surface, leading to a phase change, as depicted in Fig. 1.4. This process encompasses several heat transfer zones and distinct flow patterns, each reflecting the progressive intensification of boiling [3]:



**Fig. 1.4:** The schematic of heat transfer zones and associated flow patterns in a channel while flow boiling. Adopted from Yahyaee et al. [4] © 2019 IEEE.

- **Convective Heat Transfer of Liquid:** Prior to the onset of boiling, the heat transfer is dominated by convection in the liquid phase. Here, the liquid flows smoothly over the surface, absorbing heat without any phase change.
- **Subcooled Boiling/Bubbly Flow:** As the surface temperature rises, isolated bubbles begin to form and detach from the heated surface. These bubbles enhance mixing and energy transfer, disrupting the thermal boundary layer and improving the overall heat transfer coefficient.
- **Saturated Nucleate Boiling/Slug Flow:** With further heat input, the bubbly flow transitions into slug flow. Large vapor slugs emerge, interspersed by liquid regions. The alternating slugs of vapor and liquid create a dynamic environment for heat transfer, maintaining high efficiency.
- **Forced Convection Heat Transfer Through Liquid Film/Annular Flow:** As boiling continues, the flow evolves into an annular pattern where a

## 1.1. Boiling Phenomena and Stability Challenges

stable vapor core is surrounded by a thin liquid film along the channel walls. The heat transfer in this regime is primarily due to the liquid film's evaporation on the heated surface.

- **Liquid Deficient Region/Drop Flow:** At the highest heat fluxes, the liquid film may deplete, leading to a fully developed vapor flow interspersed with droplets, indicating a near-complete phase change from liquid to vapor. This complete phase transition indicates a significant shift in heat transfer characteristics, as the efficient liquid-phase heat transfer mechanisms are no longer present.

To ensure clarity, it is essential to note that the methods and solvers presented in this thesis are relevant and applicable to both pool and flow boiling scenarios. Also, the limitations outlined in the subsequent section may arise in either of these boiling categories, or in some cases, both. The manifestation of these limitations is highly dependent on the specific conditions and parameters of the study.

### 1.1.3 Limitations, And Stability

Boiling systems are inherently complex, encompassing various heat transfer zones and flow patterns. Adding to this complexity are several instabilities associated with boiling phenomena, which can lead to overheating and other undesirable outcomes.

During boiling, as a liquid is heated to its boiling point, it begins to vaporize at specific nucleation sites on the heated surface. This phase change from liquid to vapor involves a considerable expansion in volume. This volumetric expansion is substantial because the specific volume of a substance in its gaseous state is much larger than in its liquid state, often by orders of magnitude. This sudden increase in volume can lead to significantly higher flow velocities and create large pressure gradients within the system. Such rapid expansion and the ensuing changes in flow and pressure conditions can make the system more susceptible to instabilities. To underscore the significance of these instability issues in boiling patterns, a brief overview is provided below.

- **Rapid Bubble Growth Instability (RBGI)** is a type of instability prevalent in flow boiling systems, particularly those with small channel diameters or confined spaces. It arises when vapor bubbles grow at an accelerated rate, expanding to fill the entire cross-sectional area of the channel or a significant portion of the boiling surface in pool boiling. In flow boiling, this rapid growth occurs due to the rate of bubble formation exceeding the rate of liquid replenishment, disrupting the normal boiling process and potentially causing efficiency losses or even system damage. In

confined spaces during pool boiling, similar rapid bubble growth can lead to localized overheating and efficiency issues. The proportionally larger bubbles in these confined spaces are more likely to occupy the entire cross-sectional area, hindering the liquid flow and disrupting the heat transfer process [5].

- **Ledinegg Instability:** Named after Martin Ledinegg, who first identified this phenomenon in 1938, Ledinegg Instability is a specific concern in flow boiling systems. It results in a significant reduction in flow velocity. This instability occurs when the pressure drop versus mass flux curve for the pumping system has a shallower gradient than the demand curve for the channel. Such a discrepancy can cause an abrupt transition to a lower flow rate, disrupting the normal operation of the system. This sudden decrease in flow rate can lead to localized regions of overheating or undercooling, posing a risk to the system's stability and efficiency. For further details and comprehensive studies on this instability, references to works by Falsetti [6] and Kingston [7] are recommended.
- **Parallel Channel Instability:** This instability is predominantly relevant in flow boiling systems but can also be a consideration in complex pool boiling configurations involving interconnected channels or compartments. It develops when parallel channels, sharing identical input and output manifolds, are interconnected. Triggered by shifts in the mass flow rate supply curve or thermal influences (such as conductive contact at the inlet or exit channels), parallel channel instability can lead to significant performance issues. In flow boiling systems, fluctuations in flow or heat input conditions can result in an uneven distribution of coolant across the channels. Similarly, in pool boiling setups with interconnected channels, variations in heat input can affect the boiling behavior in different sections. This uneven distribution or boiling behavior can cause reduced coolant flow or uneven boiling in some channels, leading to increased local temperatures and potentially unsafe operating conditions. Such instability can lead to decreased system efficiency and, in extreme cases, pose safety risks due to local overheating [8].
- **Upstream Compressible Volume Instability:** This type of instability typically arises in flow boiling systems but can have implications in pool boiling under certain conditions. It occurs when a significant amount of non-condensable gas is trapped upstream of a channel, creating compressible regions. These regions can lead to pressure and density wave oscillations, especially when there are variations in the input pressure. Such oscillations can disrupt the regular fluid flow and heat transfer processes, causing instabilities that adversely affect thermal manage-

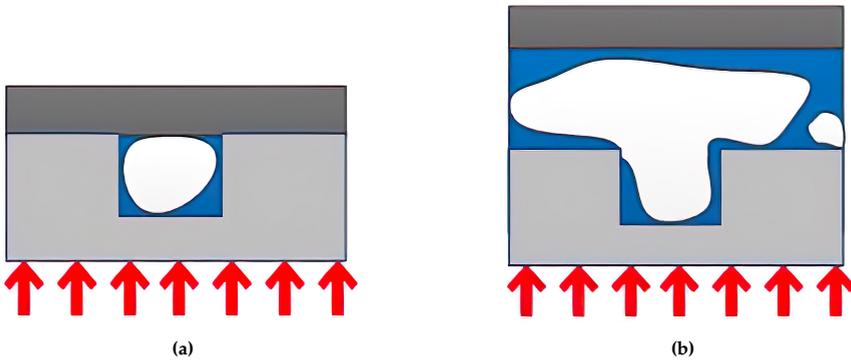
## 1.1. Boiling Phenomena and Stability Challenges

ment in both flow and, to a lesser extent, pool boiling systems. The resulting fluctuations in heat transfer rates can compromise the performance and reliability of the system [5].

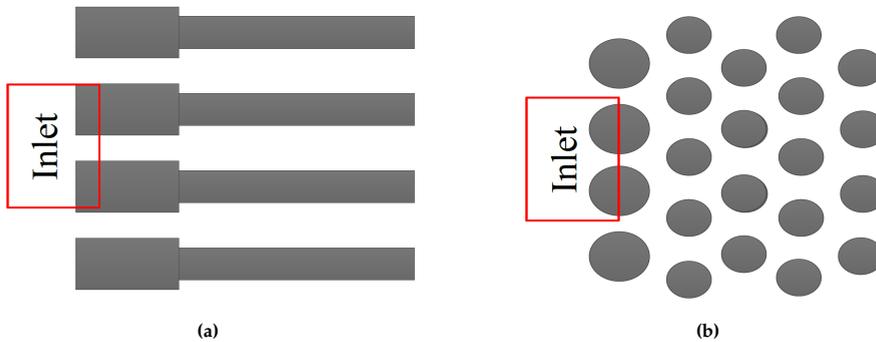
- **Critical Heat Flux Condition (CHF):** The term *critical heat flux* denotes the maximum heat flux that can be sustained by a boiling system before a significant deterioration in heat transfer occurs. This threshold is reached when the heat flux is so high that the liquid near the heated surface rapidly evaporates, leading to the formation of a vapor layer that impedes efficient heat transfer. In such a scenario, known as the 'boiling crisis' or 'departure from nucleate boiling (DNB)', the system's heat transfer capability drastically reduces due to the lower heat transfer coefficient of gas compared to liquid. CHF is a critical parameter in both pool and flow boiling systems and is essential for ensuring the reliability and safety of various thermal management applications, including power generation and electronic cooling systems. Exceeding the CHF limit can lead to overheating and potential thermal damage to the equipment [9].

The adverse effects of instabilities can be mitigated, and heat dissipation can be improved by modifying the microchannel architecture and base fluid characteristics. The following methods illustrate how these modifications can be incorporated into a variety of designs:

- **Open microchannels:** Closed-channel microchannels (conventional design) have a top plate that seals the microchannels, as shown in Fig. 1.5a. When using an open microchannel design, as seen in Fig. 1.5b, the microchannel is separated from the top plate by a gap. Due to the addition of corners to the geometry and the fact that corners are areas for bubble formation, this adjustment can enhance nucleation sites, boost heat transfer surface area, aid in vapor removal without an excessive pressure drop, decrease downstream flow resistance, and limit flow reversal. This trend hinders fast bubble formation and upstream compressible volume instability [8, 10].
- **Adding restrictors to inlet area:** Inlet restrictors can be added to rectangular channels (Fig. 1.6a) or pin fin designs (Fig. 1.6b). Cubical inlet restrictors, an extra row of larger-diameter circular pin fins, or a small nozzle or orifice are positioned in front of each channel in the inlet area. Due to the increased entrance velocity and the tendency of bubbles to travel toward the unrestricted end, this adjustment helps to eliminate vapor blockage. Rapid bubble formation instability can be dampened as a consequence of this process. Due to an increase in inlet pressure, this modification is also helpful in facing Ledinegg instability [11].



**Fig. 1.5:** Illustration of bubble growth in a) Closed and b) Open microchannels. Red arrows indicate the direction of heat flux. The blue areas represent liquid and the white areas represent the vapor created during boiling. Adapted from Yahyaee et al. [4] © 2019 IEEE.



**Fig. 1.6:** Illustration of inlet restrictors in a) Rectangular channels and b) Pin fin design. The flow enters from the left and exits on the right in both designs. Adapted from Yahyaee et al. [4] © 2019 IEEE.

- Diverging cross-section channels: Channels with stepped (Fig. 1.7a) and smooth (Fig. 1.7b) diverging cross-sections improve the downstream thrust force, due to the inverse proportion between bubble radius and surface tension force. When a slug forms in a channel with diverging geometry, the slug's diameter is smaller in the inlet area than in the outlet region. This results in a greater surface tension force at the inlet, which in turn drives the bubble toward the outlet. Moreover, this layout provides a more steady stream and a reduced pressure drop [12].
- Adding pin fins: Different pin fin designs, such as circular (Fig. 1.8a), square (Fig. 1.8b), diamond (Fig. 1.8c), and streamline pins (Fig. 1.8d), can be incorporated as an alternative to continuous rectangular channels. These pin-shaped fins help form interconnected channels. They

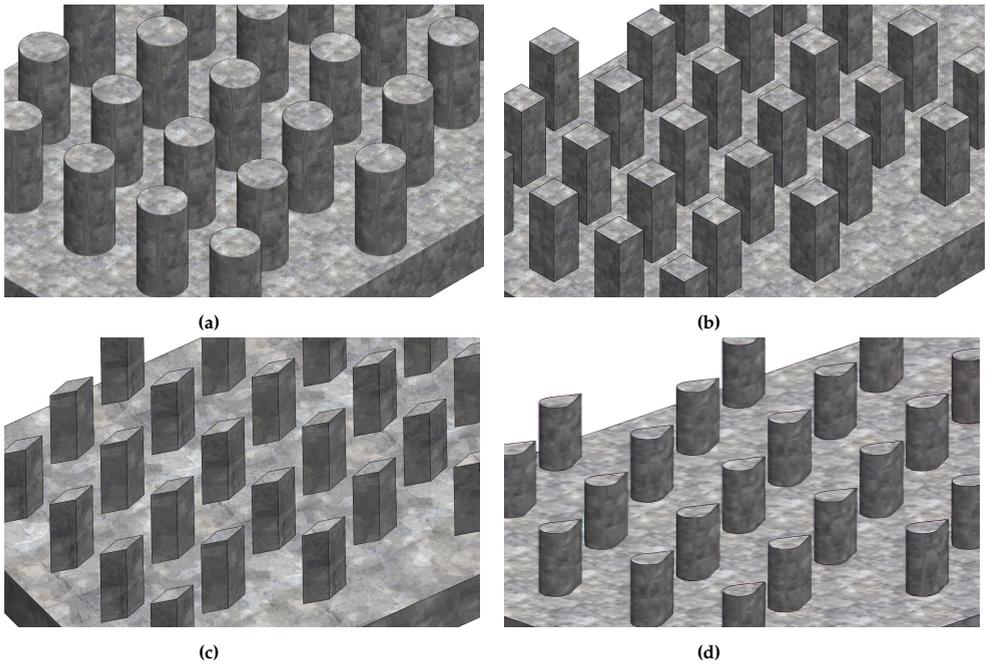
## 1.1. Boiling Phenomena and Stability Challenges



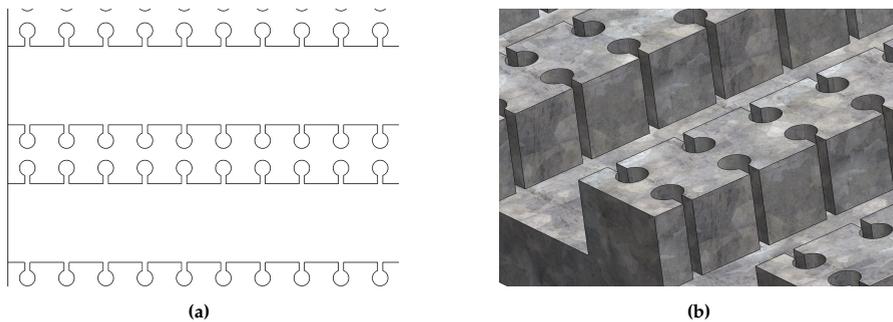
**Fig. 1.7:** Illustration of a microchannel featuring a) Stepped diverging and b) Smooth diverging cross-section. The flow enters from the left and exits on the right in both designs. Adopted from Yahyaee et al. [4] © 2019 IEEE.

allow the slug bubble to extend into primary and secondary channels while preventing flow reversal due to the interconnected passages. Furthermore, these designs increase nucleation sites due to the additional corners, reduce pressure oscillations because of the presence of primary and secondary channels, and improve heat transfer by periodically breaking up thermal boundary layers [13, 14].

- **Adding cavities:** The cavities in such designs are cube-, triangle-, or circle-shaped and serve as vapor traps, as shown in Fig. 1.9. This strategy decreases initial wall superheat, increases nucleation sites, and increases heat transfer during pool boiling. Such structures entrap large radius bubbles, in contrast to the relatively small nuclei found in natural scratches, and significantly lowering the theoretical superheat required for nucleation. Furthermore, compared to flat surfaces, it provides a substantially larger surface area for microlayer evaporation [15].
- **Application of Nanofluids:** The use of nanofluids in boiling systems has been shown to significantly improve thermal performance [16, 17]. One key advantage of nanofluids is their ability to increase the Critical Heat Flux (CHF), a crucial limit in both pool and flow boiling systems. Studies indicate that nanoparticles, when added to the boiling fluid, tend to settle on the boiling surface and improve its wettability, which helps delay the onset of CHF [18, 19]. This is because better wettability helps maintain more liquid contact with the surface, enhancing cooling. Even in film boiling, where heat transfer is usually less efficient due to a vapor layer acting as an insulator, nanofluids have shown potential to transfer heat more effectively [20]. Nanofluids consist of tiny particles, on the scale of nanometers, suspended in a base fluid. These particles can alter the fluid's heat transfer properties, like increasing thermal conductivity, though they might also reduce its specific heat capacity.



**Fig. 1.8:** Illustration of a microchannel featuring a) Circular, b) Square, c) Diamond, and d) Streamline pin fins. Adopted from Yahyaee et al. [4] © 2019 IEEE



**Fig. 1.9:** Illustration of a microchannel featuring reentrant cavities. Adopted from Yahyaee et al. [4] © 2019 IEEE.

Factors like the amount and size of the nanoparticles play a critical role in determining these effects, making the study of nanofluids a complex but promising area for enhancing the efficiency of thermal systems.

## 1.2 Gas-Liquid Interface Simulation

Nonlinear governing equations, contact line movement, and continuous deformation of the liquid-vapor interface complicate numerical solutions to thermal phase change problems drastically. Locating the position of the interface between two phases is one of the most challenging parts of modeling such processes. Due to strong property gradients, it is difficult to find interface spots on a developing interface that undergoes large deformations and topological changes. Multiple two-phase modeling classes are shown in Fig. 1.10. As shown in the diagram, there are two interface description approaches: interface capturing and interface tracking. These methods and their classes will be briefly explained further.

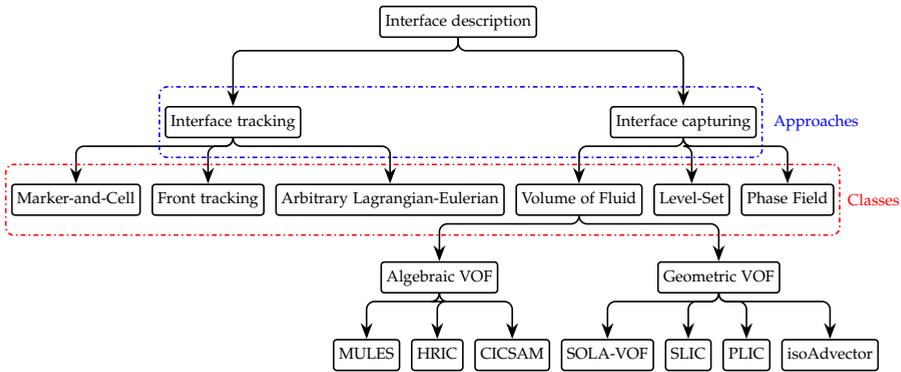


Fig. 1.10: Numerical techniques used for characterizing the interface between two distinct fluid phases.

The first approach depicted in Fig. 1.10 is interface tracking. This section offers a brief overview of three well-known types of interface tracking methods: Marker-and-Cell (MAC), Arbitrary Lagrangian-Eulerian (ALE), and Front Tracking.

Originated by Francis H. Harlow in 1965 [21], the MAC method employs markers to track the interface. These markers are dispersed throughout the study region and move with the fluid flow, enabling precise boundary tracking. In MAC, the fluid velocity field is calculated on a stationary grid, while the markers representing the boundary adapt to the fluid's movement. However, while the MAC method effectively tracks boundaries, it may encounter difficulties in scenarios involving extensive boundary changes, such as merging or breaking apart bubbles [22]. Additionally, the MAC method, despite its precision, can be computationally demanding due to the need to track numerous markers, particularly in intricate flows.

In 1970, Charles W. Hirt and his team introduced the ALE method [23].

Unlike MAC, ALE utilizes a deformable computational grid that can partially move and reshape itself along with the fluid flow, providing a more accurate representation of the boundary. The equations governing fluid flow are solved on this deformable grid, and the boundary is tracked using markers that follow the fluid's motion. While the ALE method can handle larger boundary alterations compared to MAC, it may still encounter challenges when faced with extreme changes in the boundary.

Developed by S. O. Unverdi and G. Tryggvason in 1992 [24], the Front Tracking method directly tracks the boundary. It considers the boundary as a collection of interconnected surface elements or segments. These segments move with the fluid flow, and their linkages are updated as the boundary's shape changes. Front Tracking can accurately capture the boundary's geometry and manage complex shape modifications. However, it necessitates sophisticated algorithms to maintain track of the links between surface elements when the boundary's topology changes, such as when bubbles merge or break apart. Moreover, Front Tracking is computationally intensive, particularly in 3D simulations, due to the need for continuous updates to the connections between surface elements. Despite these challenges, Front Tracking has been successfully employed in various multi-phase flow simulations.

The Marker-and-Cell, Arbitrary Lagrangian-Eulerian, and Front Tracking methods are well-established interface tracking techniques, each with its own strengths and limitations. While they provide precise geometrical information and can handle various degrees of interface deformation, they may face difficulties with substantial topological changes in the interface.

Compared to the interface tracking approach, the interface capturing approach can better handle situations involving bubble merging and collapsing. This method uses a scalar field, along with an implicit representation of phases in each cell [22]. Some of the most well-known classes of interface capture methods include VOF [25], level-set [26], and phase-field [27].

The level-set method, introduced by Stanley Osher and James Sethian in 1988 [26], uses a signed distance function to implicitly represent the interface. The zero level set of this function corresponds to the interface, while its positive and negative values denote the different fluid phases. The level-set method can handle complex interface topologies and can easily deal with merging and splitting interfaces. However, it may require additional techniques to maintain mass conservation.

The phase-field method, proposed by John W. Cahn and John E. Hilliard in 1959 [27], describes the interface using a continuous phase-field variable that smoothly transitions between the different fluid phases. This method can also handle complex interface topologies and can accurately simulate the interface dynamics, but may require a larger computational effort due to its diffuse interface representation.

The VOF method, originally developed by Charles W. Hirt and Bill D.

## 1.2. Gas-Liquid interface simulation

Nichols in 1981 [25], has become a standard in both open-source and commercial CFD software. In the VOF method, fluids are differentiated using a volume fraction field ( $\alpha$ ). Cells with  $\alpha$  values between 0 and 1 contain the interface; cells with  $\alpha = 1$  are filled with fluid 1, while cells with  $\alpha = 0$  are filled with fluid 2. This model involves solving a single system of momentum equations for both fluids, and the volume fraction of each phase in each cell is monitored throughout the domain.

In this research, the open-source C++ CFD library, OpenFOAM [28], is utilized. VOF, a prevalent method for simulating thermally induced phase change phenomena, is the pre-implemented interface capture approach in OpenFOAM. Given the use of OpenFOAM and the default existence of VOF method in it, the focus of the discussion will henceforth be on the VOF method.

The VOF method can be classified into two categories: algebraic and geometric. The algebraic VOF employs a high-resolution scheme and a compressive differencing strategy to solve the volume fraction transport equation without performing geometric calculations. Examples of algebraic VOF include the High-Resolution Interface Capturing (HRIC) [29] in STAR-CCM+, Compressive Interface Capturing Scheme for Arbitrary Meshes (CICSAM) [30] in ANSYS Fluent, and the MULES in the open-source CFD package OpenFOAM.

High-Resolution Interface Capturing (HRIC), introduced by Svend Muzaferija and Michael Perić in 1999 [29], is an algebraic VOF method used in the STAR-CCM+ CFD software package. This method provides accurate interface capturing and reduces numerical diffusion by using a high-resolution upwind scheme, which considers the direction of the flow to calculate the interface position.

Compressive Interface Capturing Scheme for Arbitrary Meshes (CICSAM), developed by O. Ubbink in 1999 [30], is an algebraic VOF method implemented in the ANSYS Fluent CFD software. CICSAM reduces numerical diffusion by employing a compressive scheme that minimizes the smearing of the fluid interface in arbitrary mesh systems. This method allows for more accurate simulations in complex geometries where traditional mesh structures may not suffice.

Multidimensional Universal Limiter for Explicit Solution (MULES), first introduced in the open-source CFD package OpenFOAM [31], is an algebraic VOF method that ensures the boundedness of the calculated volume fraction values. MULES' primary goal is to provide a stable and accurate solution to the advection equation while maintaining the conservation of mass.

Continuum-based Partial Differential Equation (PDE) discretization procedures are used to transport the volume fraction field in Algebraic VOF methods. Because of the large and rapid fluctuation in the volume fraction field across the interface, which causes interpolation and associated dis-

cretization errors when using algebraic advection methods, this approach provides challenges and may lead to complications. Furthermore, algebraic techniques suffer from a loss of numerical consistency due to artificial diffusion, making it hard for them to keep an interface's width from fluctuating. This lack of uniformity also lowers the convergence order. Recent progress in algebraic VOF approaches has helped with some of these problems, but it has not addressed them entirely [32].

For phase change problems, MULES has been widely used as the primary interface description method in OpenFOAM phase change solvers. The VOF-MULES class has a non-sharp nature, resulting in a diffused interface between phases, which can lead to inaccurate calculations of interface properties and the generation of false currents. The presence of non-physical false currents can increase interfacial mass transfer when modeling evaporation and condensation in some situations [33–36]. These issues can cause substantial numerical inaccuracies in such simulations and are considered among VOF's main drawbacks. Addressing these challenges is one of the priorities of this Ph.D. study.

The simulation of phase changes employing MULES can be improved through the adoption of auxiliary techniques like geometric VOF to avert the generation of false currents. These geometric volume fraction methodologies necessitate additional geometric operations to enhance the volume fraction, leading to a substantially more sophisticated interface compared to the algebraic method. Some of the most recognized instances of these geometric VOF strategies encompass SOLA-VOF [25], SLIC, or Simple Line Interface Calculation [37], PLIC, also known as Piecewise Linear Interface Calculation [38], and isoAdvector [39].

The SOLA-VOF (Solution Algorithm-Volume of Fluid) method was introduced by Hirt and Nichols in 1981 [25]. It is one of the earliest geometric VOF methods and uses an explicit interface tracking technique based on a volume fraction field. The method works well for problems with relatively simple interface topologies but may struggle with more complex geometries.

The Simple Line Interface Calculation (SLIC) method was proposed by Noh and Woodward in 1976 [37]. SLIC approximates the interface as a straight line within each cell, allowing for a simple and efficient interface representation. However, the accuracy of SLIC can be limited in cases with highly curved interfaces.

The Piecewise Linear Interface Calculation (PLIC) method was developed by Youngs in 1982 [38]. PLIC improves upon the SLIC method by reconstructing the interface as a piecewise linear approximation within each cell, providing better accuracy for curved interfaces. This method can handle more complex interface topologies but is more computationally demanding than SLIC.

Among the several geometric VOF methods currently in use, the isoAd-

vector method - developed by Roenby et al. in 2016 [39] - stands out due to its positive results [40] and its implementation in the OpenFOAM framework. The isoAdvector, a geometric VOF technique, is flexible enough to operate on both structured and unstructured meshes without imposing restrictions on cell shapes. A variety of research [39–44] using the isoAdvector method has been conducted, which consistently indicate that it minimizes non-physical flows near the interface.

As illustrated in Fig. 1.10, VOF is not the only method available for capturing interfaces; alternatives such as the level-set method also exist. Additionally, there are further subclasses of VOF besides MULES, such as isoAdvector. The accuracy of results obtained from thermal phase change solvers can be improved by employing these alternative approaches, including CLSVOF (combined level set with VOF) and isoAdvector methods. These methods, along with their benefits and drawbacks, are explored and discussed throughout this Ph.D. work. By investigating and comparing these methods, a more comprehensive understanding of their applications and limitations can be achieved, ultimately contributing to the development of more accurate and reliable simulations for phase change phenomena.

## 1.3 Motivation

The motivation for this Ph.D. study is twofold, focusing on advancing and refining numerical methods for the analysis of boiling processes:

1. Section 1.2 underscores the complexity of accurately locating the gas-liquid interface in boiling simulations, a crucial aspect of two-phase flow modeling. In commercial and open-source CFD software, the Volume of Fluid (VOF) method is the standard approach for thermal phase change modeling. However, it occasionally presents challenges in two-phase flow simulations, such as imprecise curvature calculations and the emergence of non-physical velocities near the interface in certain cases. This research aims to enhance the accuracy of boiling heat transfer simulations by investigating and applying alternative interface capturing methods. These alternative methods have demonstrated potential in addressing the limitations associated with VOF in two-phase flow solvers, and their integration into existing thermal phase change solvers is expected to improve the fidelity and accuracy of simulations.
2. In Section 1.1.3, various instabilities encountered in boiling cooling methods and explore strategies to overcome these challenges are discussed. One such promising strategy is the use of nanofluids to enhance boiling heat transfer efficiency. Specifically, in the film boiling regime—where heat transfer rates are typically lower due to a vapor

layer acting as an insulator—nanofluids have shown the potential to improve heat transfer. Although film boiling is critical in applications requiring high heat transfer, there is a noticeable gap in comprehensive CFD studies on nanofluid boiling, particularly in comparison to experimental research. This lack of detailed CFD studies is mainly due to the complexity involved in modeling the behavior of nanofluids during boiling. Most current CFD models either do not fully incorporate the equations needed to simulate nanoparticle concentration or limit their application only to the vapor phase. This study aims to address the existing research gap by introducing a model capable of describing the variations and distributions of nanoparticle concentrations in both the liquid and vapor phases of nanofluids, as well as at their interface during boiling. This approach enables a more detailed and accurate depiction of nanofluid film boiling, thereby enhancing our understanding and simulation capabilities of this complex phenomenon.

## 1.4 Research Questions, Hypotheses, And Objectives

Motivated by the identified challenges and prospects, this research addresses the following key questions:

1. How does substituting the MULES with isoAdvector influence the accuracy, convergence rate, and computational efficiency in boiling simulations?
2. What are the effects of integrating the level-set method with the VOF approach on the accuracy, convergence rate, and computational time efficiency of boiling simulation performance?
3. How can a governing equation for nanoparticle concentration be developed and implemented in CFD simulations to accurately represent boiling behavior in nanofluid, vapor, and interface domains?
4. What are the impacts of introducing nanoparticles on the thermophysical properties distribution and thermal efficiency during film boiling of base fluids?

Correspondingly, the research hypothesizes that:

1. Employing isoAdvector for interface description will enhance simulation performance in boiling simulations compared to the MULES approach.

#### 1.4. Research Questions, Hypotheses, and Objectives

2. The integration of the level-set method with VOF is likely to improve curvature estimation, leading to more precise boiling simulations.
3. A specialized governing equation for nanoparticle concentration will provide a more faithful representation of nanofluid boiling, potentially altering the fluid dynamics and heat transfer outcomes.
4. The utilization of nanofluids is expected to enhance flow dynamics and thermal efficiency in film boiling processes compared to pure fluids.

The objectives set forth for this Ph.D. project are:

1. Develop new thermal phase change solvers that separately incorporate isoAdvector and CLSVOF interface description methods for boiling simulations:

It is important to note that part of this study's objective is to examine the CLSVOF and isoAdvector methods against the VOF method in boiling simulations, to discern their merits and demerits. This necessitates conducting simulations in OpenFOAM. While CLSVOF and isoAdvector are well-known and have been extensively studied by other researchers, most of these studies have been conducted using in-house developed solvers, and these solvers are often not publicly available. When these solvers are available, they typically focus on two-phase scenarios and lack the capability to simulate thermal phase change. Therefore, for the investigative purposes of this study, it was essential to develop thermal phase change solvers incorporating CLSVOF and isoAdvector as interface capturing methods, marking a preliminary step in this research endeavor.

2. Evaluate the performance of isoAdvector and CLSVOF methods in boiling simulations using well-known benchmark cases:

Since new solvers were developed in the first stage, the second step will include applying this new solver to the simulation of boiling phenomena and demonstrating how the implemented interface description methods (isoAdvection and CLSVOF) influence the simulation performance. A variety of well-known thermal phase change benchmark cases are employed for this purpose.

3. To derive and implement a governing equation for nanoparticle concentration that is applicable within the liquid nanofluid, vapor, and interfacial regions during the boiling of nanofluids:

Recognizing the absence of a comprehensive model, this work will develop an equation that encapsulates the behavior of nanoparticle concentration during the boiling process. This equation will serve as a cor-

nerstone for a solver designed to accurately simulate nanofluid boiling dynamics.

4. To evaluate the influence of nanofluid application on improving the performance of film boiling using the custom-developed OpenFOAM solver:

The final objective involves using the governing equation for nanoparticle concentration, developed in the previous step, to create and implement a specialized solver in OpenFOAM for nanofluid boiling simulations. This solver will then be applied to analyze the thermal performance of nanofluids in 2D axisymmetric film boiling on a vertical cylinder. The objective is to illuminate the potential benefits nanofluids offer in enhancing heat transfer during boiling processes.

## 1.5 Thesis Outline

This Ph.D. thesis is structured as a monograph that presents a comprehensive exploration of the research conducted during the Ph.D. program. The monograph is organized into five chapters, each of which delves into specific aspects of the research:

- **Chapter 1: Introduction**

This chapter provides the background of the research, lays out the research questions, and states the objectives and significance of the study.

- **Chapter 2: Interface Description in Boiling Simulations - VOF-isoAdvector**

Chapter 2 focuses on the evaluation of the isoAdvector interface description technique during boiling simulations. This chapter examines the effects of isoAdvector on simulation results compared to VOF-MULES, using well-known benchmark cases to showcase diverse condensation and boiling conditions.

- **Chapter 3: Interface Description in Boiling Simulations - CLSVOF**

In Chapter 3, the CLSVOF interface description technique is discussed and its performance during boiling simulations is evaluated compared to VOF. Similar to Chapter 2, this chapter uses benchmark cases to highlight the advantages and disadvantages of the CLSVOF method.

- **Chapter 4: Nanofluid Film Boiling**

This chapter introduces an OpenFOAM-based solver employing the Continuous-Species-Transfer (CST) method for simulating nanoparticle transfers between vapor and liquid phases during boiling, including

## 1.6. List of Publications

governing equations for thermophoresis and Brownian motion. Focusing on film boiling, where an insulating vapor film forms between the nanofluid and the heated surface, the study conducts a two-dimensional axisymmetric film boiling analysis on a vertical cylinder.

- **Chapter 5: Conclusion**

The concluding chapter synthesizes the main contributions of the research, discusses the implications of the findings, and suggests avenues for future research.

## 1.6 List Of Publications

The research outcomes during the Ph.D. study have been disseminated in the forms of publications: two conference papers, and two journal articles (two published a) from highly regarded peer-reviewed journals, as listed in the following:

- Article A: Ali Yahyae, Amir Sajjad Bahman, Jakob Hærvig, and Henrik Sørensen. A Review: New Designs of Heat Sinks for Flow Boiling Cooling. In 2019 25th International Workshop on Thermal Investigations of ICs and Systems (THERMINIC), pages 1–6. IEEE, 2019 [4].
- Article B: Ali Yahyae, Jakob Hærvig, Amir Sajjad Bahman, and Henrik Sørensen. Numerical Simulation of Boiling in a Cavity. In 2020 26th International Workshop on Thermal Investigations of ICs and Systems (THERMINIC), pages 1–5. IEEE, 2020 [45].
- Article C: Ali Yahyae, Amir Sajjad Bahman, and Henrik Sørensen. A Benchmark Evaluation of the isoAdvection Interface Description Method for Thermally-Driven Phase Change Simulation. *Nanomaterials*, 12(10):1665, 2022 [46].
- Article D: Ali Yahyae, Amir Sajjad Bahman, Klaus Olesen, and Henrik Sørensen. Level-Set Interface Description Approach for Thermal Phase Change of Nanofluids. *Nanomaterials*, 12(13):2228, 2022 [47].
- Article E: Ali Yahyae and Henrik Sørensen. Nanoparticle Migration in Nanofluid Film Boiling: A Numerical Analysis Using the Continuous-Species-Transfer Method. *International Journal of Heat and Mass Transfer* (Under review).

## References

- [1] R. Cardenas, *Submerged jet impingement boiling thermal management*. Oregon State University, 2011.
- [2] A. Faghri and Y. Zhang, "BOILING," *Transport Phenomena in Multiphase Systems*, pp. 765–852, 1 2006.
- [3] W. M. Rohsenow, J. P. Hartnett, and Y. I. Cho, *Handbook of heat transfer*. McGraw-Hill New York, 1998, vol. 3.
- [4] A. Y. Nujukambari, A. S. Bahman, J. Hærvig, and H. Sørensen, "A Review: New Designs of Heat Sinks for Flow Boiling Cooling," in *2019 25th International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*. IEEE, 2019, pp. 1–6.
- [5] T. A. Kingston, J. A. Weibel, and S. V. Garimella, "High-frequency thermal-fluidic characterization of dynamic microchannel flow boiling instabilities: Part 2—Impact of operating conditions on instability type and severity," *International Journal of Multiphase Flow*, vol. 106, pp. 189–201, 2018.
- [6] C. Falsetti, M. Magnini, and J. R. Thome, "A new flow pattern-based boiling heat transfer model for micro-pin fin evaporators," *International Journal of Heat and Mass Transfer*, vol. 122, pp. 967–982, 2018. [Online]. Available: <https://doi.org/10.1016/j.ijheatmasstransfer.2018.02.050>
- [7] T. A. Kingston, J. A. Weibel, and S. V. Garimella, "Ledinegg instability-induced temperature excursion between thermally isolated, heated parallel microchannels," *International Journal of Heat and Mass Transfer*, vol. 132, pp. 550–556, 2019.
- [8] S. G. Kandlikar, T. Widger, A. Kalani, and V. Mejjia, "Enhanced flow boiling over open microchannels with uniform and tapered gap manifolds," *Journal of Heat Transfer*, vol. 135, no. 6, p. 61401, 2013.
- [9] S. G. Kandlikar, "History, Advances, and Challenges in Liquid Flow and Flow Boiling Heat Transfer in Microchannels: A Critical Review," *Journal of Heat Transfer*, vol. 134, no. 3, p. 034001, 2012.
- [10] L. Yin, P. Jiang, R. Xu, W. T. Wang, and L. Jia, "Visualization of flow patterns and bubble behavior during flow boiling in open microchannels," *International Communications in Heat and Mass Transfer*, vol. 85, no. May, pp. 131–138, 2017. [Online]. Available: <http://dx.doi.org/10.1016/j.icheatmasstransfer.2017.05.008>
- [11] T. Zhang, T. Tong, J.-Y. Chang, Y. Peles, R. Prasher, M. K. Jensen, J. T. Wen, and P. Phelan, "Ledinegg instability in microchannels," *International Journal of Heat and Mass Transfer*, vol. 52, no. 25-26, pp. 5661–5674, 2009.
- [12] Y. K. Prajapati, M. Pathak, M. K. Khan, and M. Kaleem Khan, "A comparative study of flow boiling heat transfer in three different configurations of microchannels," *International Journal of Heat and Mass Transfer*, vol. 85, pp. 711–722, 2015. [Online]. Available: <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2015.02.016>
- [13] Y. T. Jia, G. D. Xia, L. X. Zong, D. D. Ma, and Y. X. Tang, "A comparative study of experimental flow boiling heat transfer and pressure drop characteristics in

## References

- porous-wall microchannel heat sink," *International Journal of Heat and Mass Transfer*, vol. 127, pp. 818–833, 2018.
- [14] W. Wan, D. Deng, Q. Huang, T. Zeng, and Y. Huang, "Experimental study and optimization of pin fin shapes in flow boiling of micro pin fin heat sinks," *Applied Thermal Engineering*, vol. 114, pp. 436–449, 2017. [Online]. Available: <http://dx.doi.org/10.1016/j.applthermaleng.2016.11.182>
- [15] A. Koşar, C.-J. J. Kuo, and Y. Peles, "Boiling heat transfer in rectangular microchannels with reentrant cavities," *International Journal of Heat and Mass Transfer*, vol. 48, no. 23-24, pp. 4867–4886, 2005.
- [16] M. Dadhich, O. S. Prajapati, and V. Sharma, "A systematic review on the heat transfer investigation of the flow boiling process," 8 2021.
- [17] S. K. Das, N. Putra, and W. Roetzel, "Pool boiling characteristics of nano-fluids," *International journal of heat and mass transfer*, vol. 46, no. 5, pp. 851–862, 2003.
- [18] C. Gerardi, J. Buongiorno, L. w. Hu, and T. Mckrell, "Infrared thermometry study of nanofluid pool boiling phenomena," *Nanoscale Research Letters*, vol. 6, no. 1, 2011.
- [19] H. S. Ahn, H. J. Jo, S. H. Kang, and M. H. Kim, "Effect of liquid spreading due to nano/microstructures on the critical heat flux during pool boiling," *Applied Physics Letters*, vol. 98, no. 7, p. 071908, 2011.
- [20] A. Avramenko, I. V. Shevchuk, N. P. Dmitrenko, A. A. Moskalenko, and P. N. Logvinenko, "Unsteady convective heat transfer in nanofluids at instantaneous transition to film boiling," *International Journal of Thermal Sciences*, vol. 164, p. 106873, 2021.
- [21] F. H. Harlow and J. E. Welch, "Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface," *The physics of fluids*, vol. 8, no. 12, pp. 2182–2189, 1965.
- [22] A. Smolianski, *Numerical modeling of two-fluid interfacial flows*. University of Jyväskylä, 2001.
- [23] C. W. Hirt, "An arbitrary Lagrangian-Eulerian method for incompressible flows," in *Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics, Berkeley*, 1970.
- [24] S. O. Unverdi and G. Tryggvason, "A front-tracking method for viscous, incompressible, multi-fluid flows," *Journal of computational physics*, vol. 100, no. 1, pp. 25–37, 1992.
- [25] C. W. Hirt and B. D. Nichols, "Volume of fluid (VOF) method for the dynamics of free boundaries," *Journal of computational physics*, vol. 39, no. 1, pp. 201–225, 1981.
- [26] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations," *Journal of computational physics*, vol. 79, no. 1, pp. 12–49, 1988.
- [27] J. W. Cahn, "Free energy of a nonuniform system. II. Thermodynamic basis," *The Journal of chemical physics*, vol. 30, no. 5, pp. 1121–1124, 1959.

## References

- [28] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, "A tensorial approach to computational continuum mechanics using object-oriented techniques," *Computers in physics*, vol. 12, no. 6, pp. 620–631, 1998.
- [29] S. Muzaferija, "A two-fluid Navier-Stokes solver to simulate water entry," in *Proceedings of 22nd symposium on naval architecture*, 1999. National Academy Press, 1999, pp. 638–651.
- [30] O. Ubbink and R. I. Issa, "A method for capturing sharp fluid interfaces on arbitrary meshes," *Journal of computational physics*, vol. 153, no. 1, pp. 26–50, 1999.
- [31] H. Jasak, "Dynamic mesh handling in OpenFOAM," in *47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, 2009, p. 341.
- [32] T. Marić, D. B. Kothe, and D. Bothe, "Unstructured un-split geometrical Volume-of-Fluid methods – A review," *Journal of Computational Physics*, vol. 420, p. 109695, 11 2020.
- [33] S. Hardt and F. Wondra, "Evaporation model for interfacial flows based on a continuum-field representation of the source terms," *Journal of Computational Physics*, vol. 227, no. 11, pp. 5871–5895, 2008.
- [34] Y. Sato and B. Ničeno, "A sharp-interface phase change model for a mass-conservative interface tracking method," *Journal of Computational Physics*, vol. 249, pp. 127–161, 2013.
- [35] T. Abadie, J. Aubin, and D. Legendre, "On the combined effects of surface tension force calculation and interface advection on spurious currents within Volume of Fluid and Level Set frameworks," *Journal of Computational Physics*, vol. 297, pp. 611–636, 9 2015.
- [36] N. Samkhaniani and M. R. Ansari, "Numerical simulation of bubble condensation using CF-VOF," *Progress in Nuclear Energy*, vol. 89, pp. 120–131, 2016. [Online]. Available: <http://dx.doi.org/10.1016/j.pnucene.2016.02.004>
- [37] W. F. Noh and P. Woodward, "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*. Springer, 1976, pp. 330–340.
- [38] D. L. Youngs, "Time-dependent multi-material flow with large fluid distortion," *Numerical methods for fluid dynamics*, 1982.
- [39] J. Roenby, H. Bredmose, and H. Jasak, "A computational method for sharp interface advection," *Royal Society Open Science*, vol. 3, no. 11, p. 160405, 2016.
- [40] L. Gamet, M. Scala, J. Roenby, H. Scheufler, and J. L. Pierson, "Validation of volume-of-fluid OpenFOAM® isoAdvector solvers using single bubble benchmarks," *Computers and Fluids*, vol. 213, p. 104722, 12 2020. [Online]. Available: <https://doi.org/10.1016/j.compfluid.2020.104722>
- [41] J. Roenby, H. Bredmose, and H. Jasak, "IsoAdvector: Geometric VOF on general meshes," in *OpenFOAM®*. Springer, 2019, pp. 281–296.
- [42] K. Missios, N. G. Jacobsen, and J. Roenby, "Using the isoAdvector Geometric VOF Method for Interfacial Flows Through Porous Media Using the isoAdvector Geometric VOF Method for Interfacial Flows Through Porous Media," *9th Conference on Computational Methods in Marine Engineering (Marine 2021)*, no. June, 2021.

## References

- [43] V. Vukčević, J. Roenby, I. Gatin, and H. Jasak, "A sharp free surface finite volume method applied to gravity wave flows," *arXiv*, vol. M, 2018.
- [44] H. Scheufler and J. Roenby, "TwoPhaseFlow: An OpenFOAM based framework for development of two phase flow solvers," 2021. [Online]. Available: <https://github.com/DLR-RY/TwoPhaseFlow>
- [45] A. Yahyaee, J. Hærvig, A. S. Bahman, and H. Sørensen, "Numerical Simulation of Boiling in a Cavity," in *2020 26th International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*. IEEE, 2020, pp. 1–5.
- [46] A. Yahyaee, A. S. Bahman, and H. Sørensen, "A Benchmark Evaluation of the isoAdvection Interface Description Method for Thermally-Driven Phase Change Simulation," *Nanomaterials*, vol. 12, no. 10, p. 1665, 2022.
- [47] A. Yahyaee, A. S. Bahman, K. Olesen, and H. Sørensen, "Level-Set Interface Description Approach for Thermal Phase Change of Nanofluids," *Nanomaterials*, vol. 12, no. 13, p. 2228, 2022.

## References

## Chapter 2

# isoAdvectord approach to describe the gas-liquid interface

The thermal phase change solvers in OpenFOAM have conventionally employed the VOF-MULES method. While VOF-MULES has been effective in various applications, it has inherent limitations, such as the creation of smeared interfaces and the induction of spurious currents near the gas-liquid interface, which can compromise simulation accuracy.

The current chapter presents the description and the introduction of a VOF-isoAdvectord-based thermal phase change solver and then investigates the VOF-isoAdvectord approach through an examination of simulation results from five benchmark cases. These benchmark cases are selected because they are well-known and have been extensively studied in the field of thermal phase change, providing a solid foundation for comparison. Additionally, they have well-established analytical solutions or empirical correlations, which serve as a means to validate the accuracy and correctness of the solver's implementation. The benchmark cases are as follows:

- Stefan problem,
- Horizontal film condensation,
- Film condensation on a vertical plate,
- 2D film boiling, and
- Stationary 3D Spherical Bubble.

In addition to the four thermal phase change benchmark cases, this research incorporated a non-thermal benchmark, known as the ‘Stationary 3D Spherical Bubble Scenario.’ This case is vital for evaluating the capabilities of the VOF-isoAdvectord method in terms of reducing spurious, non-physical currents and accurately predicting curvature. Its inclusion is particularly pertinent due to the complexities inherent in thermal phase change scenarios, where interface motion often causes velocities significantly higher than those caused by spurious currents. Such a scenario complicates the task of distinguishing between physical and non-physical velocities. However, in the ‘Stationary 3D Spherical Bubble Scenario,’ the absence of interface motion implies that any observed velocities near the interface are likely due to spurious, non-physical currents, as the expected velocity around a stationary interface should ideally be zero. Consequently, this benchmark provides an unambiguous setting to assess the effectiveness of the VOF-isoAdvectord method in mitigating these non-physical currents.

This chapter is organized to offer an understanding of the thermal phase change solver based on the VOF-isoAdvectord method. Sections 2.1 and 2.2 briefly review the current literature related to the thermal phase change solvers and VOF-isoAdvectord technique, respectively. Next, Section 2.3 delves into the solver’s numerical aspects, discussing governing flow equations, discretization approaches, solver configurations, and relevant dimensionless numbers. Section 2.4 aims to highlight the specific limitations of the existing twoPhaseFlow solver, particularly in handling varying contact angles. The section serves to substantiate the need for the introduction of the thermal-PhaseChangeFlow solver, which successfully addresses these limitations. In Section 2.5, five benchmark tests are presented, offering an evaluation framework for the solver. These benchmarks are assessed against results from the VOF-MULES method, analytical solutions, and empirical data to validate the solver’s performance and reliability. Finally, the chapter concludes with Section 2.6, summarizing the key findings and insights gathered throughout the research.

## 2.1 Literature Review On Different Thermal Phase Change Solvers In OpenFOAM

In the specialized domain of computational fluid dynamics within OpenFOAM, a variety of solvers are available for addressing thermal phase changes. Notably, `evapVOFHardt` [1], one of the pioneering solvers in this field, specializes in two-phase flows with phase change. It is important to recognize `evapVOFHardt` as one of the foundational solvers in OpenFOAM for such purposes, employing the Hardt/Wondra model [2] to accurately simulate

## 2.1. Literature review on different thermal phase change solvers in OpenFOAM

the dynamics of phase interfaces, especially in scenarios involving evaporation and condensation processes.

Complementing this, `interThermalPhaseChangeFoam` [3] offers a versatile, VOF-based tool for thermally driven phase phenomena, adept at handling the precise simulation of thermal effects on phase transitions.

`PhaseChangeHeatFoam` [4] is another solver, specifically tailored for simulations of boiling and condensation. It features an enhanced VOF interface tracking system, augmented by the Lafaurie smoothing filter [5], crucial for accurately capturing the complex dynamics of phase interfaces in these processes.

Moreover, OpenFOAM's ESI-OpenCFD Versions provide solvers like `interCondensatingEvaporatingFoam` and `icoReactingMultiphaseInterFoam`. The former, `interCondensatingEvaporatingFoam`, focuses on incompressible, non-isothermal immiscible fluids, employing a VOF methodology for phase change between a fluid and its vapor, and is particularly effective in single-fluid phase change scenarios.

In contrast, `icoReactingMultiphaseInterFoam` caters to more complex situations involving multiple incompressible, non-isothermal immiscible fluids. This solver is suitable for intricate multiphase systems with several fluids undergoing phase transitions, ideal for simulations of reactive or interacting systems.

All these solvers utilize the VOF-MULES method for interface capturing, a critical aspect for ensuring numerical stability and accuracy in the representation of fluid interfaces in phase change simulations.

Although the VOF method is widely used, it has drawbacks, such as a tendency to blur the interface and introduce artificial flows. These issues can be mitigated by incorporating the `isoAdvector` geometric class into the VOF method.

The `isoAdvector` class, which is also part of the OpenFOAM suite, stands out as a robust geometric VOF approach, as noted in Section 1.2. Numerous studies have validated its effectiveness in minimizing spurious flows in multiphase simulations [6–11].

Specifically, for thermal phase change simulations, Scheufler et al. [11] have developed the `twoPhaseFlow` solver. This solver has proven its worth across various test scenarios and represents a significant advancement in leveraging the `isoAdvector` method for interface tracking. However, it is worth noting that `twoPhaseFlow` shows limitations, especially when handling varying contact angles, as further detailed in Section 2.4.

Given these limitations, there is a need for further development of solvers capable of more accurately simulating thermal phase changes with the `isoAdvector` method.

## 2.2 Literature Review On VOF-IsoAdvectord Method

The isoAdvectord method has gained considerable attention in the field of CFD for its capabilities in simulating interfacial flows between two incompressible fluids [9, 12]. Initially developed as a geometric VOF technique, it was integrated into OpenFOAM's official release starting from version v1706, thereby enhancing the solver named 'interIsoFoam' [9].

The algorithm was designed to overcome the limitations of algebraic VOF methods, particularly in handling complex geometries and maintaining a sharp interface between fluid phases. One of its key features is the use of a Reconstructed Distance Function (RDF) for computational interface reconstruction. This enhancement has been shown to achieve second-order convergence with reduced absolute errors across various mesh types [11].

In terms of applications, the isoAdvectord method has been extended to model flows in porous media, especially in the realms of coastal and marine engineering [8]. The method has demonstrated its ability to accurately incorporate the effects of porosity on fluid flow through the inclusion of source terms in the Navier-Stokes equations, such as Darcy-Forchheimer forces [13].

When it comes to computational efficiency, the isoAdvectord method has been found to outperform OpenFOAM's standard algebraic interfacial flow solver, 'interFoam', originally developed by Weller [14]. The method has undergone validation against both numerical and experimental benchmarks, showing excellent interface sharpness, shape conservation, and volume conservation [12].

Despite its promising features and applications, the isoAdvectord method is not without challenges. Comprehensive validation data for flows dominated by surface tension are still sparse [8]. Additionally, there is room for improvement in the method's coupling with other solution algorithms in OpenFOAM, such as the PISO-based pressure-velocity solution algorithm [12].

Overall, the isoAdvectord method has made significant contributions to the advancement of CFD simulations involving interfacial flows. Its geometric approach, computational efficiency, and adaptability to complex geometries make it a robust tool for a variety of engineering applications. However, further research is needed to comprehensively validate the method and optimize its integration with other computational algorithms.

In line with the research objectives of this study, a specialized solver named thermalPhaseChangeFlow has been developed in OpenFOAM v2006. While it inherits its core structure from the interPhaseChangeFoam solver in OpenFOAM, it further integrates elements from the phaseChangeHeatFoam [15] and interFlow solvers, originally developed for OpenFOAM 2.2 and OpenFOAM-v1706 versions, respectively. The interPhaseChangeFoam

solver is designed for simulating incompressible two-phase flows, particularly relevant in cavitation studies where phase change is predominantly driven by pressure variations. It uses a VOF framework to capture the interface between the liquid and vapor phases. The `thermalPhaseChangeFlow` solver extends these foundational capabilities by incorporating thermal phase change models and the `isoAdvector` method, aiming to improve both the accuracy and computational efficiency of interface tracking in thermal phase change simulations.

## 2.3 Numerical Formulation

### 2.3.1 Governing Equations

In the VOF approach, a unified set of governing equations is employed to model both liquid and vapour states, thereby eliminating the requirement to solve distinct equation sets for each phase. The governing equations for two distinct, incompressible fluids—mass, momentum, thermal energy, and interface advection—are detailed as follows:

The equation for mass conservation is:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0, \quad (2.1)$$

where  $\rho$  signifies fluid density and  $\mathbf{U}$  represents the velocity field.

The equation governing momentum can be expressed as:

$$\frac{\partial(\rho \mathbf{U})}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) - \nabla \cdot (\mu(\nabla \mathbf{U}^T + \nabla \mathbf{U})) = -\nabla P + \rho \mathbf{g} + \sigma \kappa \nabla \alpha. \quad (2.2)$$

In this equation,  $\mu$  indicates dynamic viscosity,  $P$  is the pressure, and  $\mathbf{g}$  is the acceleration due to gravity. The term  $\sigma \kappa \nabla \alpha$  is introduced to accommodate surface tension effects, utilizing the Continuous Surface Force (CSF) model [16]. Here,  $\sigma$  is the surface tension coefficient, while  $\kappa$  defines the mean curvature of the interface, determined by [17]:

$$\kappa = -\nabla \cdot \left( \frac{\nabla \alpha}{|\nabla \alpha|} \right), \quad (2.3)$$

where  $\alpha$  symbolizes the volume proportion of the liquid phase (equation (2.6)), and  $\nabla \alpha$  is computed utilizing the linear Gauss technique, which is readily available in OpenFOAM.

For thermal energy conservation, the equation is:

$$\frac{\partial(\rho c_p T)}{\partial t} + \nabla \cdot (\rho c_p \mathbf{U} T) = \nabla \cdot (k \nabla T) - \dot{m}''' (h_v - h_l), \quad (2.4)$$

where  $c_p$  denotes the heat capacity at constant pressure,  $T$  shows temperature,  $k$  is the thermal conductivity,  $\dot{m}'''$  is the transferred mass flux rate which can be calculated using the models presented in section 2.3.2, and  $h_v$  and  $h_l$  are the vapour and liquid enthalpies, respectively.

For interface advection, the following equation is derived [15]:

$$\frac{\partial \alpha}{\partial t} + \mathbf{U} \cdot \nabla \alpha = -\dot{m}''' \left[ \frac{1}{\rho_l} - \alpha \left( \frac{1}{\rho_l} - \frac{1}{\rho_v} \right) \right], \quad (2.5)$$

where  $\rho_l$  and  $\rho_v$  are liquid and vapour densities.

In the VOF approach, the variable of volume fraction ( $\alpha$ ) is employed to demarcate between differing phases, and their connecting interface is mathematically described by equation (2.6):

$$\alpha = \frac{V_l}{V_{\text{total}}} = \begin{cases} 1 & \text{in the liquid domain} \\ 0 < \alpha < 1 & \text{at the interface} \\ 0 & \text{in the vapor domain} \end{cases}, \quad (2.6)$$

In this equation,  $\alpha$  symbolizes the liquid volume fraction,  $V_l$  is the volume of the cell occupied by the liquid, and  $V_{\text{total}}$  is the overall cell volume. A cell entirely filled with liquid corresponds to  $\alpha = 1$ , while a cell completely in vapor form is represented by  $\alpha = 0$ . The transitional interface between these extreme states is delineated by  $0 < \alpha < 1$ .

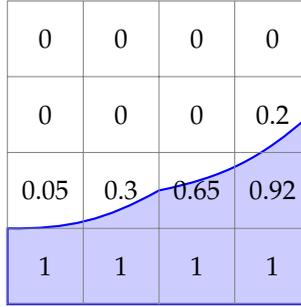
Fig. 2.1 illustrates the VOF method in action, displaying how it represents the interface between two different phases within a computational grid. The blue region corresponds to the liquid phase with an  $\alpha$  value of 1, whereas the white region is indicative of the vapour phase with an  $\alpha$  value of 0. The curve that intersects these regions designates the interface, indicating an  $\alpha$  range between 0 and 1. This illustration aids in clarifying the function of the VOF approach and how the volume fraction variable differentiates the liquid, vapour, and their intervening interface.

The VOF technique employs the volume fraction  $\alpha$  for dual purposes: it serves to demarcate the interface and to compute composite fluid properties. These include mixture density  $\rho_{\text{mix}}$ , mixture viscosity  $\mu_{\text{mix}}$ , specific heat at constant pressure for the mixture  $c_{p,\text{mix}}$ , and mixture thermal conductivity  $k_{\text{mix}}$ , as shown in equations 2.7, 2.8, 2.9, and 2.10. These composite properties are derived as volume fraction-weighted averages of the corresponding properties in the liquid  $\rho_l, \mu_l, c_{p,l}, k_l$  and vapour  $\rho_v, \mu_v, c_{p,v}, k_v$  phases.

$$\rho_{\text{mix}} = \alpha \rho_l + (1 - \alpha) \rho_v, \quad (2.7)$$

$$\mu_{\text{mix}} = \alpha \mu_l + (1 - \alpha) \mu_v, \quad (2.8)$$

### 2.3. Numerical Formulation



**Fig. 2.1:** Depiction of how the VOF method identifies liquid and vapor phases. The blue-shaded cells represent liquid with a full liquid volume fraction ( $\alpha$ ) of 1. The white cells indicate vapor with a liquid volume fraction ( $\alpha$ ) of 0. The curve shows the interface between the liquid and vapor, where the liquid volume fraction ( $\alpha$ ) is between 0 and 1, indicating a mix of both phases.

$$c_{p,\text{mix}} = \alpha c_{p,l} + (1 - \alpha)c_{p,v}, \quad (2.9)$$

$$k_{\text{mix}} = \alpha k_l + (1 - \alpha)k_v, \quad (2.10)$$

In CFD, three essential criteria are conservativeness, convergence, and boundedness. **Conservativeness** pertains to the principle that the numerical methods must conserve key physical quantities, such as mass, momentum, and energy, within the computational domain. In the context of multiphase flow simulations, particularly, conserving the mass of each phase is vital to maintain the accuracy and stability of the simulation.

**Convergence** is the second critical aspect and refers to the approximation of the numerical solution to the exact solution as the computational mesh is refined. This criterion is essential for the reliability and validity of the numerical findings, as it ensures that the solution becomes more accurate with finer computational grids.

**Boundedness**, the third criterion, ensures that the numerical solutions remain within physically plausible ranges. For example, in the VOF methodology, the volume fraction  $\alpha$  must be bounded between 0 and 1. This constraint is necessary to prevent computational instabilities and to maintain physical realism in the simulation.

Addressing the criteria of conservativeness, convergence, and boundedness, the VOF technique introduces a divergence term in the equation for  $\alpha$  advection. This term plays a crucial role in differentiating between homogeneous liquid and vapor regions, where it remains zero, and the interface region, where its significance increases. The term's purpose is to maintain consistency with the physical properties of these phases and to accurately capture the transition between them.

This divergence term is an integral part of the modified advection equation for the VOF method, denoted as equation (2.11). Its inclusion is pivotal in maintaining the integrity of the interface between phases, ensuring that the VOF method adheres to the key principles of conservativeness, convergence, and boundedness in CFD simulations [14].

The modified advection equation for the volume fraction  $\alpha$  is pivotal in this discussion, as shown below [18]:

$$\frac{\partial \alpha}{\partial t} + \mathbf{u} \cdot \nabla \alpha + \nabla \cdot (\alpha(1 - \alpha)\mathbf{u}_c) = -\dot{m}''' \left[ \frac{1}{\rho_l} - \alpha \left( \frac{1}{\rho_l} - \frac{1}{\rho_v} \right) \right], \quad (2.11)$$

Here,  $\mathbf{u}_c$  represents the compressive velocity, crucial for minimizing dispersion errors, and is computed normal to the interface [19]:

$$\mathbf{u}_c = \min(C_\alpha |U|, \max |U|) \frac{\nabla \alpha}{|\nabla \alpha|}, \quad (2.12)$$

The parameter  $C_\alpha$ , serving as a compression-enhancing factor, typically ranges between 1 and 4 to effectively limit the spread of the interface [20].

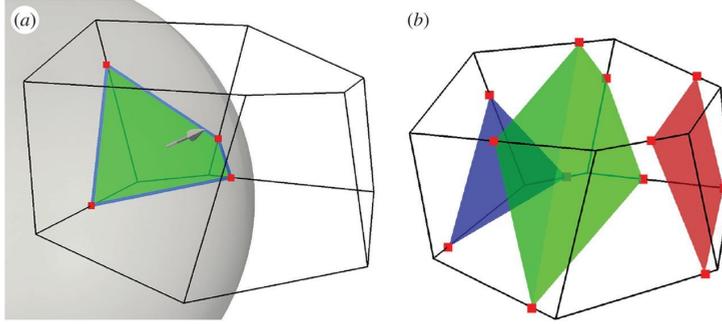
Transitioning from the specific implementation of the VOF method, it is important to consider the broader context of interface advection techniques in CFD. Two prominent methods in this domain are MULES and isoAdvectord. While they share similarities in their fundamental approach, these methods exhibit distinct differences in their treatment of the 'interface advection step.' Both MULES and isoAdvectord utilize the PIMPLE algorithm for solving the coupled pressure-velocity equations. This algorithm is an iterative method that amalgamates elements of both the SIMPLE and PISO algorithms, which are widely recognized for their efficiency in handling the complex dynamics of fluid flow.

IsoAdvectord, emerging as an innovative alternative to MULES, leverages the concept of isosurfaces for more accurate face flux calculations at the interface [21]. Distinct from MULES, isoAdvectord adopts a geometrical strategy, focusing on the dynamic reshaping of isosurfaces in each interface cell [9]. This method offers enhanced precision in capturing the fluid interface, especially beneficial in complex multiphase flow scenarios.

The technique involves constructing a unique isosurface within each interface cell for every computational step, as illustrated in Fig. 2.2(a). This approach facilitates a more accurate representation of the interface's intersection with the cell faces, leading to precise calculations of face fluxes, as depicted in Fig. 2.2(b).

While MULES utilizes a traditional flux correction strategy, focusing on maintaining boundedness and conservativeness in the volume fraction field, isoAdvectord's geometrical reconstruction of isosurfaces offers a sharp, accurate representation of the fluid interface. This innovative approach sig-

### 2.3. Numerical Formulation



**Fig. 2.2:** (a) Sectional plane cutting through a computational cell, marked by intersection points on the cell face, referred to as the isosurface. (b) Depiction of the isosurface's successive positions at three distinct intermediate times  $\tau$  within one time step. Adapted from [9].

nificantly reduces the smearing effect commonly encountered in traditional methods, rendering isoAdvector particularly effective in simulations where the detailed behavior of the interface is critical.

In summary, the distinction between MULES and isoAdvector is marked by their respective approaches to interface advection: MULES with its flux correction methodology and isoAdvector with its geometrical, isosurface-based strategy. This difference is pivotal in determining the most suitable method for specific CFD applications, particularly those involving complex interface dynamics.

For an in-depth understanding of the isoAdvector methodology and its underlying equations, a detailed study by Roenby et al. is suggested [9].

#### 2.3.2 Models For Thermal Phase Change

To complete the aforementioned set of equations, suitable phase change models are essential for calculating the transferred mass flux rate ( $\dot{m}'''$ ). In this solver, two mass transfer models of Lee and Tanasawa are employed.

##### Tanasawa's model

Tanasawa's model [22] is rooted in the Hertz-Knudsen equation [23], which describes mass flux across the phase boundary considering variations in temperature and pressure across the interface. The equation is as follows:

$$\dot{m}''' = \frac{2 - \gamma_c}{2} \sqrt{\frac{M}{2\pi R}} \left[ \frac{P_v}{\gamma_c \sqrt{T_v}} - \frac{P_l}{\gamma_e \sqrt{T_l}} \right], \quad (2.13)$$

Here,  $P_v$  and  $P_l$  represent the vapour and liquid pressures, respectively.  $T_v$  and  $T_l$  are the temperatures of the vapour and liquid phases, respectively.  $M$

denotes the molar mass of the substance.  $R = 8.314 \text{ J/molK}$  is the universal gas constant.  $\gamma_c$  and  $\gamma_e$  are the coefficients for condensation and evaporation, respectively. The simplified equation is:

$$\dot{m}'' = \frac{2 - \gamma}{2\gamma} \sqrt{\frac{M}{2\pi R}} \frac{\rho_v H_{1v} (T - T_{\text{sat}})}{(T_{\text{sat}})^{3/2}}, \quad (2.14)$$

In this equation,  $H_{1v}$  is the latent heat of vaporization.  $T_{\text{sat}}$  is the saturation temperature, dependent on local pressure conditions. The volumetric mass transfer rate is:

$$\dot{m}''' = \dot{m}'' |\nabla \alpha|. \quad (2.15)$$

### Lee's model

Lee [24] proposed a model assuming phase change at a constant pressure, in quasi-thermodynamic equilibrium conditions. The equations for the volumetric mass transfer rate are:

For condensation ( $T < T_{\text{sat}}$ ):

$$\dot{m}''' = r_c (1 - \alpha) \rho_v \frac{T - T_{\text{sat}}}{T_{\text{sat}}}, \quad (2.16)$$

For evaporation ( $T > T_{\text{sat}}$ ):

$$\dot{m}''' = r_e \alpha \rho_l \frac{T - T_{\text{sat}}}{T_{\text{sat}}}, \quad (2.17)$$

In these equations,  $r_c$  and  $r_e$  are empirical coefficients known as the mass transfer intensity factors, with units in  $s^{-1}$ .  $T$  is the local temperature.  $T_{\text{sat}}$  is the saturation temperature at a given pressure.

### 2.3.3 Discretization Methods And Solver Configuration

The governing equations of fluid flow are discretized using various numerical schemes provided in OpenFOAM. Specifically, these discretization schemes are detailed in the `fvSchemes` file within the OpenFOAM environment. A focus on second-order methods is maintained, as illustrated in Table 2.1.

### 2.3. Numerical Formulation

**Table 2.1:** Discretization methods employed in the solution algorithm for benchmark cases. Adopted from Yahyaee et al. [25] © 2022 Licensee MDPI, Basel, Switzerland.

Components	Methods
Time component	Backward
Momentum equation convection	vanLeerV
Energy equation convection	vanLeer
Compression velocity in momentum equation	interfaceCompression
Momentum equation diffusion	Gauss Linear corrected
Viscosity in momentum equation	Gauss Linear

For each benchmark case examined in this study, specific `fvSolution` and `controlDict` files are utilized. These files are foundational for the effective operation of the OpenFOAM solver and the control of the simulation runtime.

The `fvSolution` files set forth the algorithms and solver settings that are critical for solving the system of linear equations derived from discretizing the governing equations of fluid flow. Meanwhile, the `controlDict` files configure the runtime control parameters for the simulations, which include, among other settings, the time-step size, start and stop times, data write frequency, and the post-processing utilities to be activated.

The appendices containing these critical files for each benchmark case are as follows:

- Stefan Problem: `fvSolution` in Appendix A1.1, `controlDict` in Appendix A1.2
- Horizontal Film Condensation: `fvSolution` in Appendix A1.3, `controlDict` in Appendix A1.4
- Film Condensation on a Vertical Plate: `fvSolution` in Appendix A1.5, `controlDict` in Appendix A1.6
- 2D Film Boiling: `fvSolution` in Appendix A1.7, `controlDict` in Appendix A1.8
- Stationary 3D Spherical Bubble: `fvSolution` in Appendix A1.9, `controlDict` in Appendix A1.10

#### 2.3.4 Dimensionless Numbers

The simulation results presented in this chapter and future chapters are non-dimensionalized based on several dimensionless physical or geometrical groups. These groups include film thickness  $\delta^*$ , time  $t^*$ , temperature  $\theta^*$ ,

velocity  $U^*$ , the highest capillary number  $Ca_{\max}$ , capillary time  $t_{\sigma}^*$ , and the pressure coefficient  $C_p$ .

The non-dimensional film thickness  $\delta^*$  is defined as:

$$\delta^* = \frac{\delta}{L}, \quad (2.18)$$

where  $\delta$  denotes the actual film thickness, and  $L$  is the domain length.

The non-dimensional time  $t^*$  is:

$$t^* = \frac{t}{t_0} = \frac{t}{\sqrt{\frac{L}{g}}}, \quad (2.19)$$

with  $t$  as the real time, and  $t_0$  as the system's characteristic time scale.

Non-dimensional temperature  $\theta^*$  is defined as:

$$\theta^* = \frac{T - T_{\text{sat}}}{T_s - T_{\text{sat}}}, \quad (2.20)$$

The non-dimensional velocity  $U^*$  is:

$$U^* = \frac{U}{U_{\sigma}} = \frac{U}{\frac{\sigma}{\mu}}, \quad (2.21)$$

The highest capillary number, denoted by  $Ca_{\max}$ , is:

$$Ca_{\max} = \frac{U}{U_{\sigma}} = \frac{U}{\frac{\sigma}{\mu}}, \quad (2.22)$$

The dimensionless capillary time  $t_{\sigma}^*$  is:

$$t_{\sigma}^* = \frac{t}{t_{\sigma}} = \frac{t}{\sqrt{\frac{\rho_v D^3}{\sigma}}}, \quad (2.23)$$

The dimensionless pressure coefficient  $C_p$  is:

$$C_p = \frac{P}{\frac{\rho U_{\sigma}^2}{2}}, \quad (2.24)$$

The use of these non-dimensional groups allows for a more universal interpretation of the results, rendering them applicable to a variety of physical conditions beyond those explicitly examined in this study.

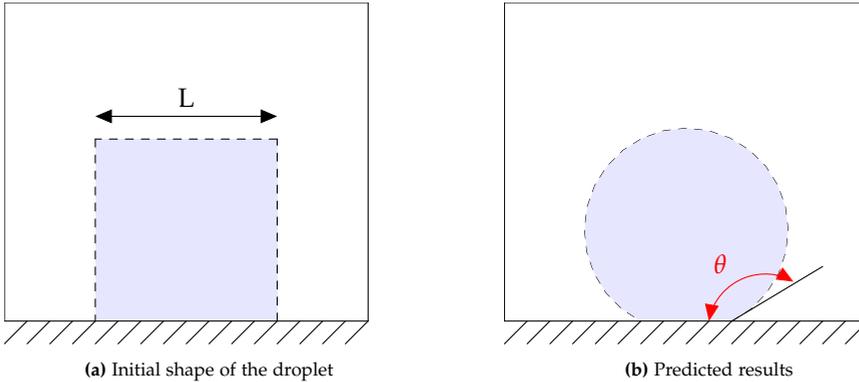
## 2.4 Comparison: ThermalPhaseChangeFlow Solver And TwoPhaseFlow Solver

Several studies have utilized the isoAdvector technique to analyze multi-phase flows [6–10]. Notably, the research by Scheufler et al. [11] introduced the ‘twoPhaseFlow’ solver, developed specifically for phase transition simulations using the isoAdvector interface characterization method. While this solver has passed various validations, certain limitations are identified under specific operational conditions, elaborated in this section.

On the other hand, a new solver named ‘thermalPhaseChangeFlow’ has been created for the present research. This solver is based on OpenFOAM’s ‘interPhaseChangeFoam’ solver, in contrast to ‘twoPhaseFlow’, which is based on the ‘icoReactingMultiphaseInterFoam’ solver. Moreover, the ‘thermalPhaseChangeFlow’ solver does not suffer from the limitations associated with varying contact angles, unlike the ‘twoPhaseFlow’ and ‘icoReactingMultiphaseInterFoam’ solvers. To highlight this, a case study is subsequently discussed. This subsection investigates a static bubble in a 2D setting, under zero-gravity conditions, on surfaces with varied wettability attributes. As depicted in Fig. 2.3a, a square volume of water—serving as a droplet—is enclosed within a rectangular domain filled with vapor. This square volume has dimensions  $L \times L$ , where  $L$  is 1 mm. Boundary conditions for this domain are primarily symmetric planes, except for the bottom boundary, which is designated as a wall. Here, different contact angles are used to examine the influence of surface wettability on the behavior of the static bubble. The computational study employs a grid structure of  $64 \times 64$  cells to ensure sufficient resolution. Results are extracted after a simulation time of 1 second. Table 2.2 shows the thermophysical properties used in this case.

**Table 2.2:** Fluid properties used in 2D static bubble case, Stefan problem and horizontal condensation benchmark cases [26].

	Dimension	Liquid	Vapour
Thermal conductivity, $k$	$\text{W m}^2 \text{K}^{-1}$	0.67337	0.03643
Density, $\rho$	$\text{kg m}^{-3}$	887.13	5.1450
Viscosity, $\mu$	$\text{Pa s}$	$1.512 \times 10^{-4}$	$1.502 \times 10^{-5}$
Specific heat capacity $c_p$	$\text{kJ kg}^{-1} \text{K}^{-1}$	2.687	2.687
Latent Heat, $h$	$\text{kJ kg}^{-1}$	762.52	2777.1
Surface tension, $\sigma$	$\text{N m}^{-1}$	0.042217	



**Fig. 2.3:** Schematic of 2D static bubble under zero gravity and without any phase change on surfaces with different wettabilities.

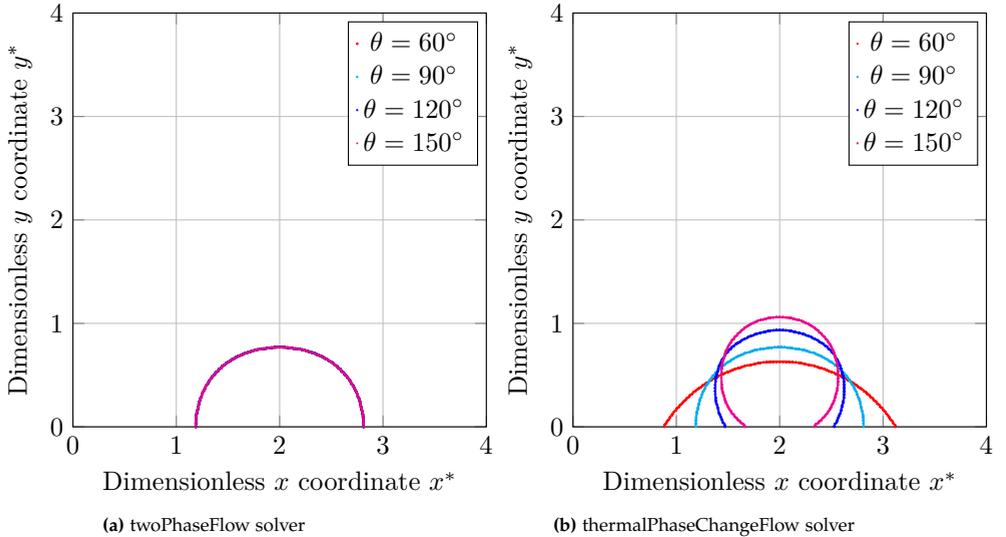
Surface tension induces a force within the bubble's outer layer, striving to reduce the surface area given the contained volume. This tendency leads theoretical simulations to forecast a spherical bubble form, as shown in Fig. 2.3b.

In surface science, the capacity of a surface to repel or attract water is generally termed as hydrophobicity or hydrophilicity. A surface is labeled hydrophobic if its static water contact angle  $\theta > 90^\circ$  and hydrophilic when  $\theta < 90^\circ$ . This investigation explores contact angles that span both hydrophobic and hydrophilic conditions, specifically  $\theta = [60^\circ, 90^\circ, 120^\circ, 150^\circ]$ .

An evaluation of 'thermalPhaseChangeFlow' and 'twoPhaseFlow' solvers was undertaken to gauge their proficiency in forecasting the ultimate shape of bubbles under different wettability conditions. Both the  $x$  and  $y$  coordinates were normalized with a scaling factor  $L$ . Fig. 2.4a revealed that the 'twoPhaseFlow' solver struggles with capturing either hydrophobic or hydrophilic surface behavior. In particular, the solver inadequately models contact angles deviating from  $90^\circ$ , effectively rendering all scenarios as though they possess a  $90^\circ$  contact angle.

Contrastingly, Fig. 2.4b showcases the 'thermalPhaseChangeFlow' solver's robustness in dealing with a spectrum of contact angles. This adaptability enhances the solver's applicability for exhaustive phase change studies involving diverse surface wettability.

## 2.5. Benchmark cases



**Fig. 2.4:** The performance of `thermalPhaseChangeFlow` and `twoPhaseFlow` solvers on predicting the final shape of the droplet with different values of contact angles  $\theta = [60^\circ, 90^\circ, 120^\circ, 150^\circ]$ .

## 2.5 Benchmark Cases

The in-house OpenFOAM thermally driven phase change solver is used to compare the results of the VOF-MULES and VOF-isoAdvector techniques. Due to the lack of a sharp interface, the  $\alpha = 0.5$  isosurface is taken to represent the vapour-liquid interface. In order to compare the outcomes of various benchmark instances, analytical solutions, and experimental correlations serve as the basis of comparison.

Tanasawa’s phase change model [22] is employed for simulations. Structured static meshes are employed for all the benchmark cases. Dynamic meshes are not utilized, partly because they are incompatible with two 1D benchmarks in the study. To maintain consistency across all benchmarks, the choice of using static meshes is deemed appropriate. Benchmark cases are examined, using a systematic grid refinement to demonstrate the difference in solvers’ reliability at varying grid sizes, in which  $N_k$  refers to the total number of grids.

Files pertinent to each benchmark case—including geometry, grid structures, and other necessary files for OpenFOAM simulations—are accessible at the following URL: <https://github.com/AAU-OpenFOAM/LSThermalPhaseChangeFlow><sup>1</sup>.

<sup>1</sup>Note that the solver is named ‘LSThermalPhaseChangeFlow’ on GitHub. This difference in naming is because the ‘thermalPhaseChangeFlow’ solver has been further developed to in-

### 2.5.1 Stefan Problem

The Stefan problem, commonly employed as a benchmark scenario for testing thermally-driven phase change phenomena in new solvers, involves evaporation due to heat conduction [26–29]. This benchmark case is graphically represented in Fig. 2.5. The domain for this problem is one-dimensional (1D), with a length of  $L = 2$  mm. In this setup, the left wall is maintained at a higher temperature ( $\Delta T_{\text{sup}} = 10$  K), leading to evaporation at the interface and causing it to move to the right. The simulation initiates with a thin layer of vapor ( $\delta = 0.08$  mm) adhering to the left wall. The thermophysical properties and characteristics used in this benchmark case are detailed in Table 2.2.

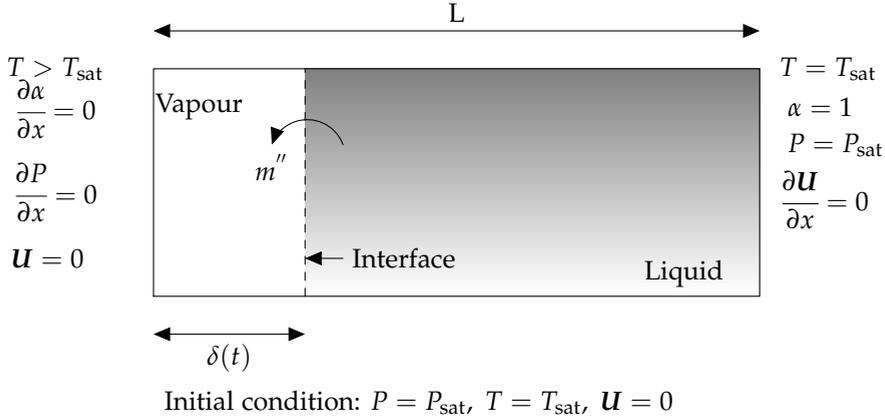


Fig. 2.5: Schematic of the Stefan problem as well as its accompanying boundary conditions. Adopted from Yahyaee et al. [25] © 2022 Licensee MDPI, Basel, Switzerland.

The domain-along temperature profile ( $T(x, t)$ ) and vapour-film thickness ( $\delta$ ) can be obtained analytically [26] as follows:

$$T(x, t) = T \Big|_{x=0} - \frac{\Delta T_{\text{sup}}}{\text{erf}(\epsilon)} \text{erf} \left( \frac{x}{2\sqrt{\frac{k_v t}{\rho_v c_{p,v}}}} \right), \quad (2.25)$$

$$\delta(t) = 2\epsilon \sqrt{\frac{k_v t}{\rho_v c_{p,v}}}, \quad (2.26)$$

clude Level Set interface description methods. However, this is a completely different subject outside the scope of this chapter. To use the solver as ‘thermalPhaseChangeFlow’, simply set the ‘interfaceMethod’ to ‘isoAdvection’ in the ‘fvSolution’ file.

## 2.5. Benchmark cases

in which the constant  $\epsilon$  being defined [26] by:

$$\epsilon \exp(\epsilon^2) \operatorname{erf}(\epsilon) = \frac{c_{p,v} \Delta T_{\text{sup}}}{(h_v - h_l) \sqrt{\pi}}. \quad (2.27)$$

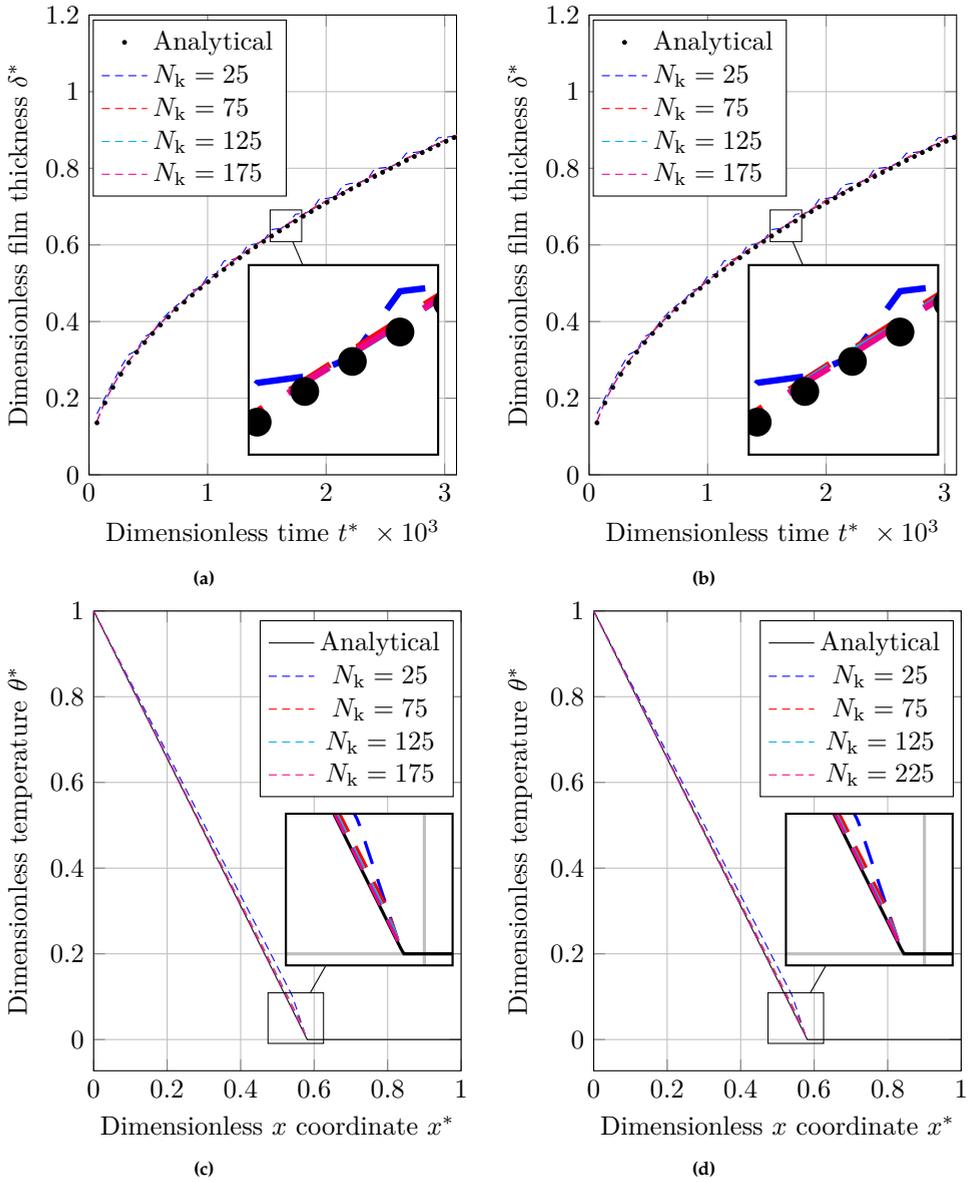
The results of solving this benchmark scenario for  $N_k = [25, 75, 125, 175]$  grid numbers are reported in Fig. 2.6 as:

- The interface's dimensionless position ( $\delta^*$ ) versus dimensionless time ( $t^*$ ), and
- The dimensionless temperature ( $\theta^*$ ) throughout the domain.

Fig. 2.6a and Fig. 2.6b illustrate that the dimensionless vapour film thickness ( $\delta^*$ ) graph exhibits a zigzag pattern when the coarsest grid architecture ( $N_k = 25$ ) is used. With denser grid layouts, the zigzag route smooths out (noises are eliminated), and graphs approach that of an analytical method.

Fig. 2.6c and Fig. 2.6d illustrate the dimensionless temperature throughout the domain at  $t = 20$  s. It is shown, coarser grid configurations ( $N_k = [25, 75]$ ) provide dimensionless temperatures that are larger than the analytical solution at the areas close to the gas-liquid interface.

From the analysis of these figures, it can be observed that for this benchmark case, under the specified conditions and with a 1D domain length of  $L = 2$  mm, a mesh size of  $1.6 \times 10^{-2}$  mm (corresponding to  $N_k = 125$ ) can be considered optimal for simulations using VOF-MULES and VOF-isoAdvector. This choice is based on the observation that results with  $N_k = 125$  exhibit better stability and fewer fluctuations compared to coarser meshes, and the outcomes are closely aligned with those from the finer mesh of  $N_k = 175$ , the next refined grid level.



**Fig. 2.6:** A comparison between the numerical results obtained using the VOF-MULES and VOF-isoAdvectord and the analytical solution for the Stefan problem. The comparisons include: the dimensionless thickness of the vapour film ( $\delta^*$ ) vs dimensionless time ( $t^*$ ), measured by (a) VOF-MULES and (b) VOF-isoAdvectord; dimensionless temperature distribution ( $\theta^*$ ) across the dimensionless length ( $x^*$ ), obtained using (c) VOF-MULES and (d) VOF-isoAdvectord. Adopted from Yahyaee et al. [30] © 2022 Licensee MDPI, Basel, Switzerland.

The aforementioned graphs do not clearly reveal the differences between

## 2.5. Benchmark cases

the VOF-MULES and VOF-isoAdvector simulations. Therefore, to highlight these differences more accurately, a logarithmic error graph is presented (Fig. 2.7). For a selection of grid sizes, the L2 norm of the error in the temperature distribution ( $e_T$ ) can be calculated using the following formula:

$$e_T = \sqrt{\frac{\sum_{i=1}^{N_k} \left( \frac{T_{\text{num},i} - T_{\text{ana},i}}{T_{\text{ana},i}} \right)^2}{N_k}}. \quad (2.28)$$

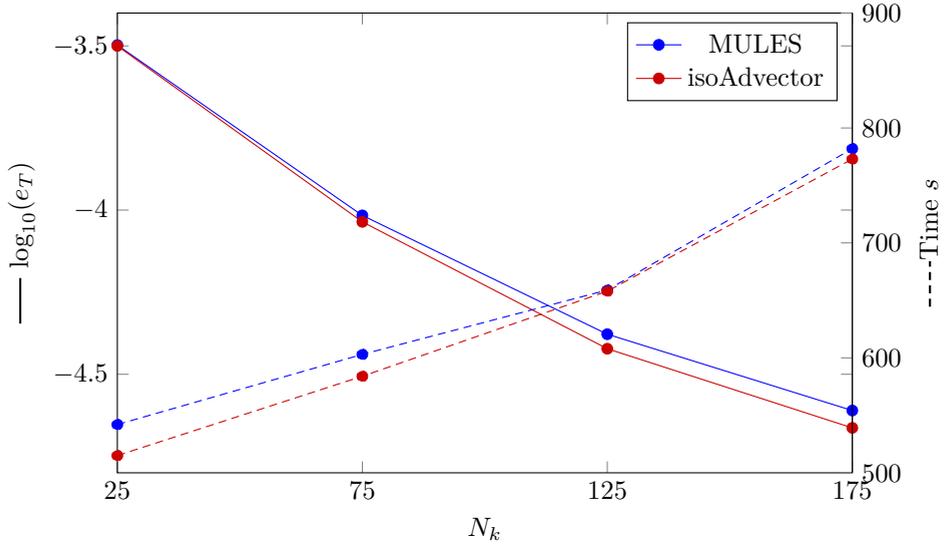
In this equation,  $T_{\text{num},i}$  is the temperature calculated by the numerical simulation at the  $i$ th grid point, and  $T_{\text{ana},i}$  is the temperature calculated by the analytical solution at the same grid point. The L2 norm of the error, denoted as  $e_T$ , provides a measure of the overall discrepancy between the numerical and analytical solutions. By summing the square of the difference at each grid point, normalizing it by the number of grid points, and taking the square root, the L2 error norm presents the error distribution into a single value. This allows for a more straightforward comparison of the accuracy between the VOF-MULES and VOF-isoAdvector solvers.

As seen in Fig. 2.7, both VOF-MULES and VOF-isoAdvector converge as the number of grids grows. The error introduced by VOF-isoAdvector is slightly less than that introduced by VOF-MULES. The studied calculation time (shown as dashed lines) for the various numbers of grids is also provided in Fig. 2.7. The results demonstrate that the VOF-isoAdvector has a faster computation time for all grid sizes.

Along with the accuracy and the computation time, convergence rate ( $R_k$ ) is also studied for each benchmark case, which is defined by:

$$R_k = \frac{\log(e_k/e_{k-1})}{\log(N_k/N_{k-1})}, \quad (2.29)$$

where  $e_k$  is the error introduced by partitioning the domain into  $N_k$  grids. In contrasting the VOF-MULES and VOF-isoAdvector methodologies, this analysis shows that the VOF-isoAdvector method exhibits a slightly higher convergence rate between the finest grid structures ( $N_k = [125, 175]$ ).



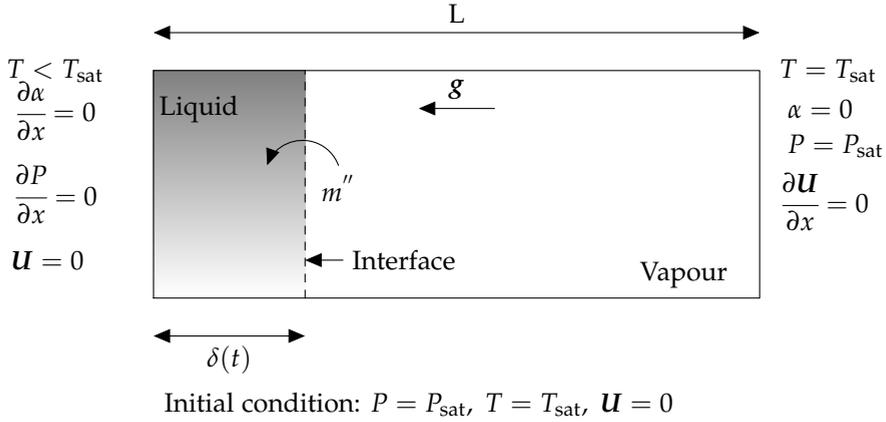
**Fig. 2.7:** Solid lines show the logarithmic representation of the error ( $\epsilon_T$ ) (equation 2.28) and dashed lines reflect computation time of VOF-MULES and VOF-isoAdvectord at various grid structures for solving Stefan problem. The horizontal axis ( $N_k$ ) shows number of grids along the  $x$  axis. Adopted from Yahyaee et al. [30] © 2022 Licensee MDPI, Basel, Switzerland.

## 2.5.2 Horizontal Film Condensation

A numerical simulation of the horizontal film condensation benchmark case [15, 31, 32] is carried out and the results are compared to Nusselt's film theory. Fig. 2.8 illustrates the test case and its boundary conditions. The free-stream with the saturation temperature is represented by the right wall, while the left wall is maintained at a subcooled temperature of  $\Delta T_{\text{sub}} = 30$  K. Upon the condensation of vapour, a thin liquid film forms on the left wall. The domain length for this 1D model is  $L = 0.3$  mm. The simulation initiates with a thin layer of liquid ( $\delta = 0.01$  mm) adhering to the left wall. The film thickness ( $\delta_{\text{ana}}(t)$ ) as a function of time is derived analytically using a control volume analysis, assuming a linear temperature profile from sub-saturation to saturation. The analytical expression for the liquid film thickness ( $\delta_{\text{ana}}(t)$ ) is given by the following equation [15]:

$$\delta_{\text{ana}}(t) = \left[ 2t \left( \frac{k_l}{\rho_l c_{p,l}} \right) \left( \frac{1}{2} + \frac{h_v - h_l}{c_{p,l} \Delta T_{\text{sub}}} \right)^{-1} \right]^{\frac{1}{2}}. \quad (2.30)$$

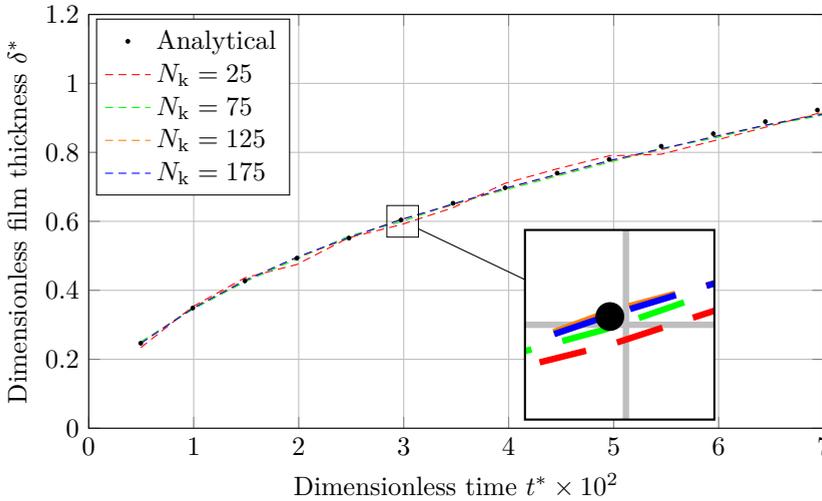
## 2.5. Benchmark cases



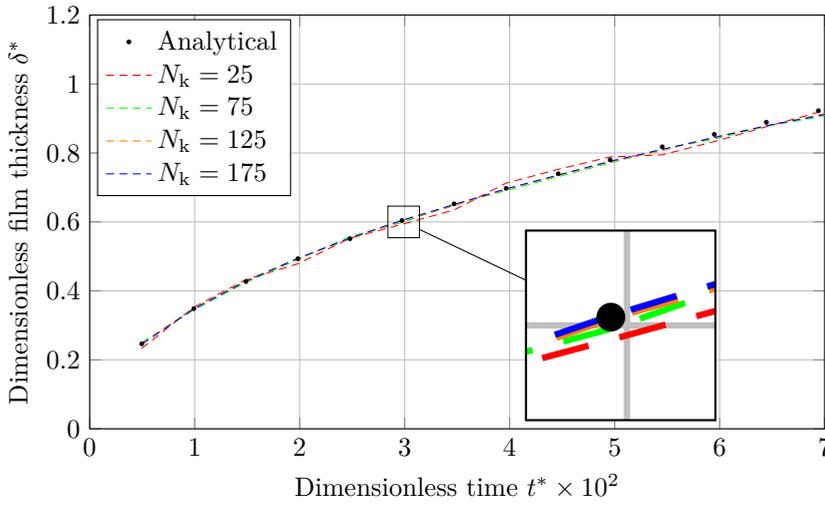
**Fig. 2.8:** Schematic of horizontal film condensation benchmark case and its boundary conditions. Adopted from Yahyaee et al. [25] © 2022 Licensee MDPI, Basel, Switzerland.

Table 2.2 displays the thermophysical parameters used in this test case. The study is done using four different numbers of grids in the  $x$  direction ( $N_k = [25, 75, 125, 175]$ ).

The development of the condensed liquid film thickness ( $\delta^*$ ) as a function of dimensionless time ( $t^*$ ) is shown graphically in Fig. 2.9, and Fig. 2.10, respectively, using the VOF-MULES and VOF-isoAdvectord methods. For all of the shown mesh sizes, the results correspond with the analytical ones.



**Fig. 2.9:** Dimensionless thickness of the liquid film ( $\delta^*$ ) over dimensionless time ( $t^*$ ) calculated using VOF-MULES in horizontal film condensation benchmark scenario. Adopted from Yahyaee et al. [30] © 2022 Licensee MDPI, Basel, Switzerland.



**Fig. 2.10:** Dimensionless thickness of the liquid film ( $\delta^*$ ) over dimensionless time ( $t^*$ ) calculated using VOF-isoAdvectord in horizontal film condensation benchmark scenario. Adopted from Yahyaee et al. [30] © 2022 Licensee MDPI, Basel, Switzerland.

Upon reviewing the graphs related to the horizontal film condensation benchmark, it becomes apparent that a grid size of  $2.4 \times 10^{-3}$  mm is optimal for simulations conducted with a 1D domain length of  $L = 0.3$  mm, corresponding to a grid count of  $N_k = 125$ . This grid size, when utilized in conjunction with VOF-MULES and VOF-isoAdvectord techniques, yields results with enhanced stability and reduced fluctuations. The performance of  $N_k = 125$  is notably superior to coarser meshes and demonstrates close agreement with the results from a finer mesh of  $N_k = 175$ . Despite the higher resolution of  $N_k = 175$ , the improvements in the simulation outcomes are marginal, suggesting that  $N_k = 125$  offers a computationally efficient yet accurate mesh configuration for this specific case study.

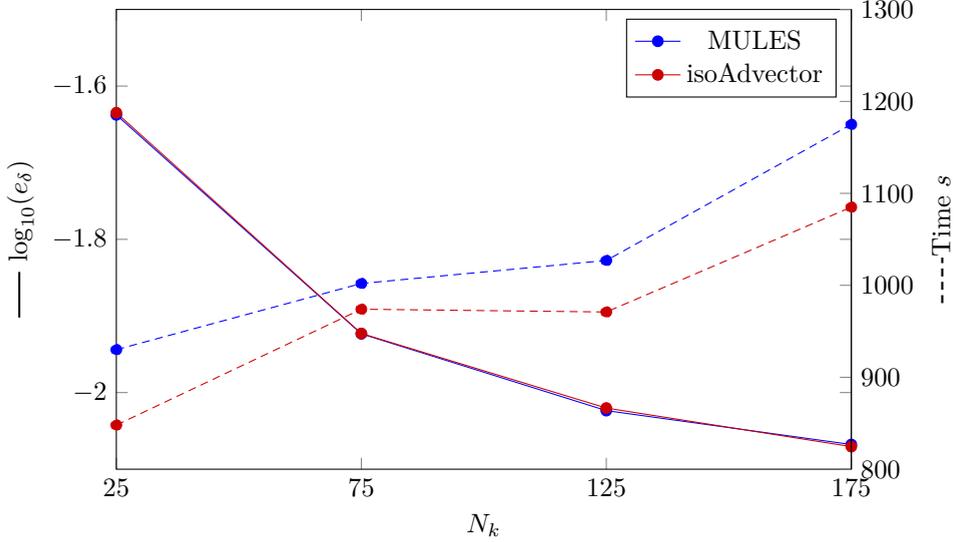
Similarly to Stefan problem, the L2 error, convergence rate and computation time analysis are done in the horizontal condensation benchmark case. The L2 error for a layer of condensed liquid is defined as:

$$e_\delta = \sqrt{\frac{\sum_{i=1}^{N_{\delta t}} \left( \frac{\delta_{\text{num},i} - \delta_{\text{ana},i}}{\delta_{\text{ana},i}} \right)^2}{N_{\delta t}}}, \quad (2.31)$$

where  $N_{\delta t}$  is the total number of time increments. Fig. 2.11 shows that VOF-MULES and VOF-isoAdvectord both provide identical results, with a similar convergence rate (equation 2.29) between the finest grid structures ( $N_k = [125, 175]$ ). Fig. 2.11 also displays the results of the analysis of computing time. It can be seen from the graph that VOF-isoAdvectord achieves an

## 2.5. Benchmark cases

improved outcome than VOF-MULES in regards to computation speed while preserving the same precision.

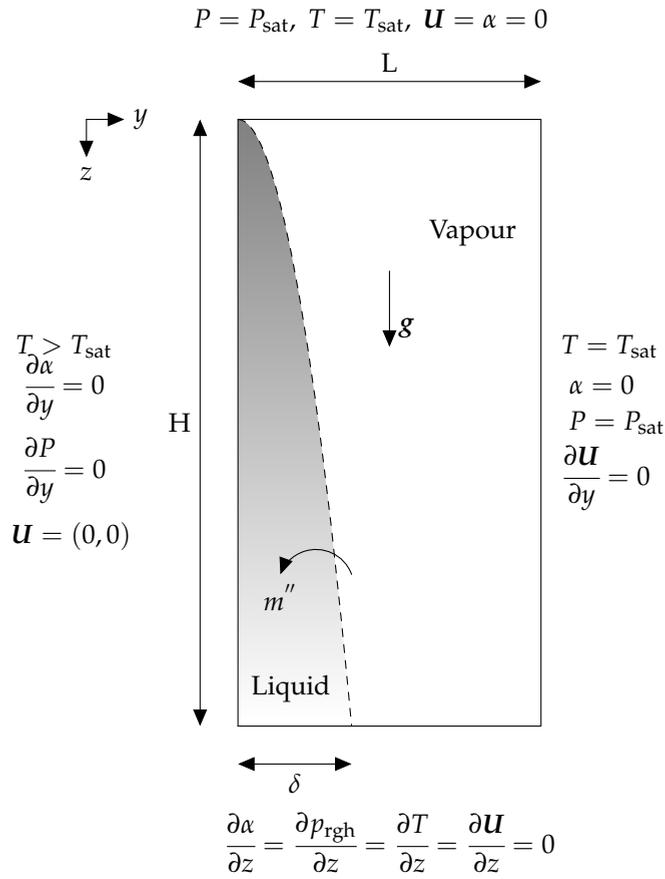


**Fig. 2.11:** Solid lines show the logarithmic representation of the error  $e_\delta$  (2.31) and dashed lines reflect computation time of VOF-MULES and VOF-isoAdvector at various grid structures for solving horizontal film condensation benchmark case. The horizontal axis ( $N_k$ ) shows the number of grids along the  $x$  axis. Adopted from Yahyae et al. [30] © 2022 Licensee MDPI, Basel, Switzerland.

### 2.5.3 Laminar Film Condensation On A Vertical Plate

The condensation of a film on a vertical plate is a well-established benchmark case [15, 31, 32], modeled here to validate against its corresponding analytical solution. The 2D domain, with dimensions  $L \times H = 0.5 \text{ mm} \times 3 \text{ mm}$ , is filled with vapour at the saturation temperature. The left wall is subcooled with a  $\Delta T_{\text{sub}} = 20 \text{ K}$  and is coated by a liquid film, as depicted in Fig. 2.12. The simulation starts with a thin layer of liquid ( $\delta = 0.005 \text{ mm}$ ) attached to the left wall.

The material characteristics pertinent to this case study are itemized in Table 2.3. To dissect this scenario, four grid configurations with  $N_k = [25 \times 50, 75 \times 150, 125 \times 250, 175 \times 350]$  are deployed.



Initial condition:  $P = P_{\text{sat}}, T = T_{\text{sat}}, \mathbf{U} = 0$

**Fig. 2.12:** Schematic of the laminar film condensation on a vertical plate case with its boundary conditions. Adopted from Yahyaee et al. [25] © 2022 Licensee MDPI, Basel, Switzerland.

## 2.5. Benchmark cases

**Table 2.3:** Fluid properties used to solve two benchmark cases of laminar film condensation on a vertical plate, and the 2D film boiling.

	<b>Dimension</b>	<b>Liquid</b>	<b>Vapour</b>
Thermal conductivity, $\lambda$	$\text{W m}^2 \text{K}^{-1}$	0.545	0.538
Density, $\rho$	$\text{kg m}^{-3}$	402.4	242.7
Viscosity, $\mu$	$\text{Pa s}$	$4.67 \times 10^{-5}$	$3.23 \times 10^{-6}$
Specific heat capacity $C_p$	$\text{kJ kg}^{-1} \text{K}^{-1}$	218	352
Latent Heat, $h$	$\text{kJ kg}^{-1}$	1963.5	2240
Surface tension, $\sigma$	$\text{N m}^{-1}$	$7 \times 10^{-5}$	

There are two assumptions needed for the analytical solution to be found for this problem [15, 32]:

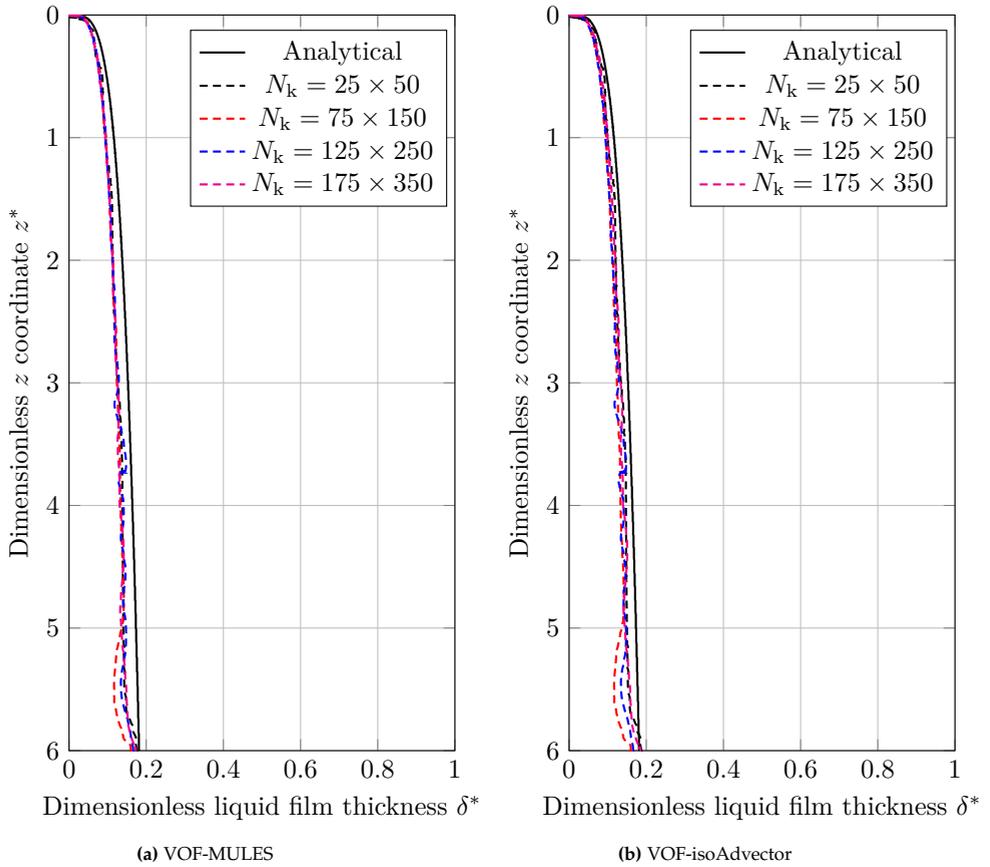
- There is a linear temperature profile over the film, and
- Interfacial shear stress and inertial forces are disregarded.

The equation 2.32 presents the analytical solution which describes the relationship between the film thickness ( $\delta$ ) and the vertical coordinate  $z$  [15]:

$$\delta = \left[ \frac{4\mu_l k_l \Delta T_{\text{sub}} z}{g(h_v - h_l)\rho_l(\rho_l - \rho_v)} \right]^{\frac{1}{4}}, \quad (2.32)$$

where  $\mu_l$  is the dynamic viscosity of the liquid phase,  $k_l$  is the thermal conductivity of the liquid phase,  $\Delta T_{\text{sub}}$  is the temperature below the saturation temperature,  $g$  is the acceleration due to gravity,  $h_v$  and  $h_l$  are the specific enthalpies of the vapour and liquid phases, respectively,  $\rho_l$  is the density of the liquid phase, and  $\rho_v$  is the density of the vapour phase. The  $z$  coordinate is used to denote the vertical position in the flow, and the thickness of the film,  $\delta$ , varies as a function of  $z$ .

Fig. 2.13a and 2.13b provide the VOF-MULES and VOF-isoAdvector results for the condensed liquid film thickness, respectively. The convergence to the analytical results occurs with grid refinement.



**Fig. 2.13:** A representation of the condensed liquid film interface, obtained by numerical (VOF-MULES and VOF-isoAdvecter) and analytical methods for laminar condensation on a vertical plate benchmark case and with different grid structures ( $N_k = [25 \times 50, 75 \times 150, 125 \times 250, 175 \times 350]$ ). the axes have been made dimensionless by scaling the values with respect to the length  $L$ .

In evaluating the mesh for laminar film condensation on a vertical plate, a grid size of  $125 \times 250$  provides a well-balanced approach between precision and computational efficiency. OpenFOAM's non-uniform grid employs a 'simpleGrading' parameter set to  $(4, 0.25, 1)$ . This grid strategy is essential for achieving a refined mesh near the wall cooled below the fluid's boiling point, which is a region characterized by steep temperature gradients and the formation of a thin liquid film.

The 'simpleGrading' value of 0.25 indicates that the cell size reduces to one quarter of its previous size from the top boundary moving downward, ensuring a higher resolution where the physical effects are critical. The value of 1 in the 'simpleGrading' parameter refers to the  $z$ -direction, which, in a 2D simulation, does not affect the mesh since there is no depth variation.

## 2.5. Benchmark cases

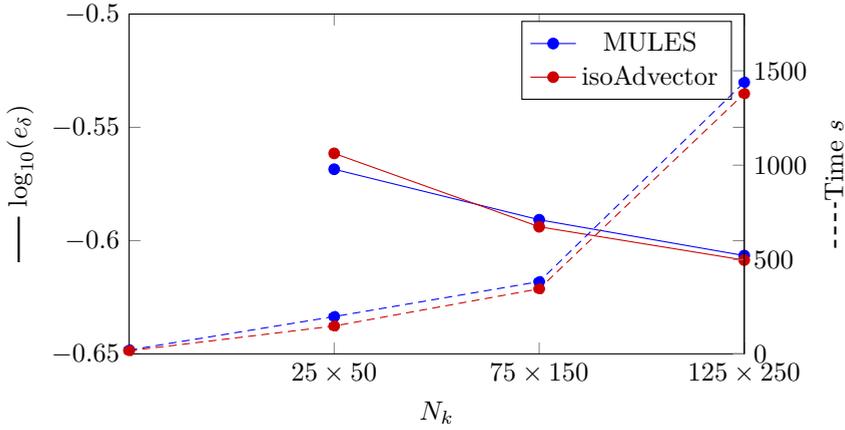
The factor of 4 suggests that the cell size expands by a factor of four moving away from the left wall, thus allocating computational resources efficiently by using larger cells where detailed resolution is less critical.

This mesh configuration aligns well with Nusselt's analytical film theory and is identified as the optimal choice for simulating vertical plate condensation. While a finer grid such as  $175 \times 350$  offers increased detail, it does not significantly improve upon the results obtained with the  $125 \times 250$  mesh, thereby affirming the latter as a judicious grid selection for this study.

To evaluate the VOF-isoAdvector and VOF-MULES methods in the 2D vertical film condensation problem, the L2 error was calculated and is shown in Fig. 2.14. The L2 error for the condensed liquid layer thickness in this case is defined by the equation:

$$e_\delta = \sqrt{\frac{\sum_{i=1}^{N_k} \left( \frac{\delta_{\text{num},i} - \delta_{\text{ana},i}}{\delta_{\text{ana},i}} \right)^2}{N_k}}, \quad (2.33)$$

where  $N_k$  is the total number of grids along  $y$ . The outcomes from the  $25 \times 50$  grid setup are not included in the  $e_\delta$  assessment or shown in Fig. 2.14 due to the significant interface instability and the development of an excessively broad and dispersed interface in this particular configuration. As displayed in Fig. 2.14, the error associated with the VOF-isoAdvector method closely aligns with that of the VOF-MULES, indicating comparable convergence rates for both approaches. Additionally, dashed lines in Fig. 2.14 provide insight into the computational time analysis, revealing that the VOF-isoAdvector algorithm performs computations more swiftly than its VOF-MULES counterpart.



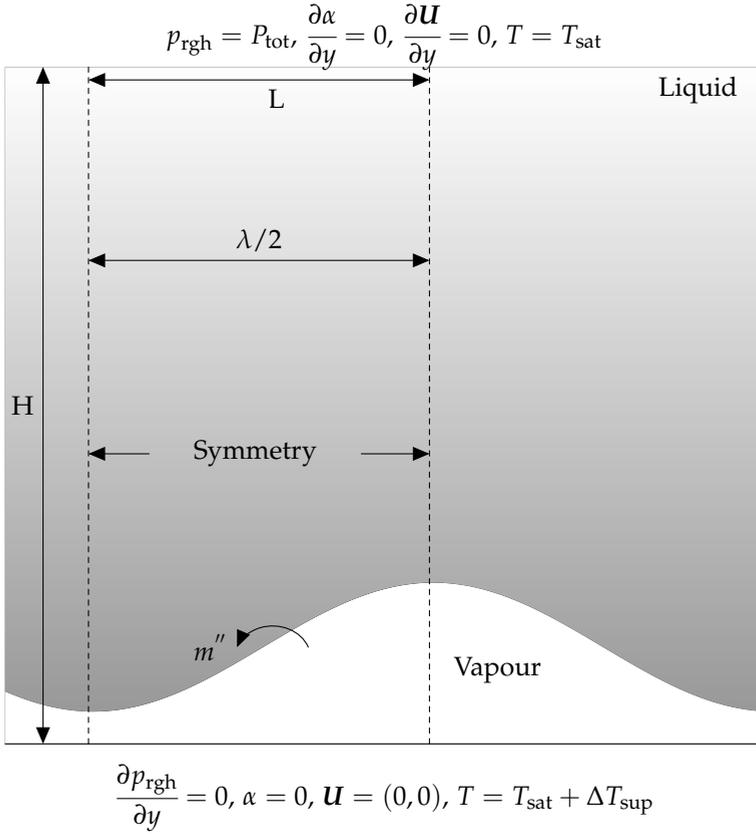
**Fig. 2.14:** Solid lines show the logarithmic representation of the error  $e_\delta$  (equation 2.31) and dashed lines reflect the computation time of VOF-MULES and VOF-isoAdvectord at various grid structures for solving laminar film condensation on a vertical plate benchmark case. The horizontal axis ( $N_k$ ) shows the number of grids. Adopted from Yahyaee et al. [30] © 2022 Licensee MDPI, Basel, Switzerland.

## 2.5.4 2D Film Boiling

In this well-known benchmark case [33–37], between the surface and the saturated liquid is a very thin layer of vapour in the form of a sinusoidal wave (see Fig. 2.15). The liquid is at saturation temperature, and the surface is slightly warmer ( $\Delta T_{\text{sup}} = 5$  K). The domain as well as its accompanying boundary conditions are shown in Fig. 2.15. For this benchmark case a characteristic length ( $\lambda$ ) is defined as the equation below [38]:

$$\lambda = \sqrt{\frac{\sigma}{(\rho_l - \rho_v)g}}. \quad (2.34)$$

## 2.5. Benchmark cases



**Fig. 2.15:** Geometry and the associated boundary conditions for the 2D film boiling benchmark case. Adopted from Yahyaee et al. [25] © 2022 Licensee MDPI, Basel, Switzerland.

The studied domain size is  $\lambda/2 \times \lambda = 1.15 \times 2.3 \text{ mm}^2$ . The simulation starts with an initial vapour film attached to the bottom wall. The thickness of this vapour film is defined by [39]:

$$\delta = \frac{\lambda_0}{128} \left( 4 + \cos \left( \frac{2\pi x}{\lambda_0} \right) \right), \quad (2.35)$$

where  $\lambda_0$  is the Taylor equation's critical wavelength as [39]:

$$\lambda_0 = 2\pi \sqrt{\frac{3\sigma}{(\rho_l - \rho_v)g}}. \quad (2.36)$$

In this situation, the Nusselt number can be calculated by [39]:

$$\text{Nu} = \frac{\int_0^L \left( \frac{\lambda}{\Delta T} \frac{\partial T}{\partial y} \Big|_{y=0} \right) dx}{L}. \quad (2.37)$$

The correlation introduced by Berenson [38] is utilized to predict the Nusselt number for this case as:

$$\text{Nu} = 0.425 \left[ \frac{\rho_v(\rho_l - \rho_v)g(h_v - h_l)}{k_v\mu_v\Delta T} \right] \quad (2.38)$$

To study this scenario, four grid configurations with  $N_k = [150 \times 300, 175 \times 350, 200 \times 400, 225 \times 450]$  are chosen. Table 2.3 displays the thermophysical properties used for solving this case. The  $x$  and  $y$  coordinates are scaled with  $\lambda$  (equation 2.34) and  $t$  is scaled with  $t_0$  (equation 2.19) to get the dimensionless values.

In the context of this specific benchmark case, the focus is not on quantifying the error between the simulation results and the reference data. This approach is justified because the reference data is derived from experimental correlations rather than exact analytical solutions. This correlation is subject to variations stemming from different conditions and boundary settings, making it less than ideal for precise error quantification. Instead, the correlation-based results serve as a qualitative benchmark to ascertain that the simulation outputs lie within a reasonable range of expected outcomes.

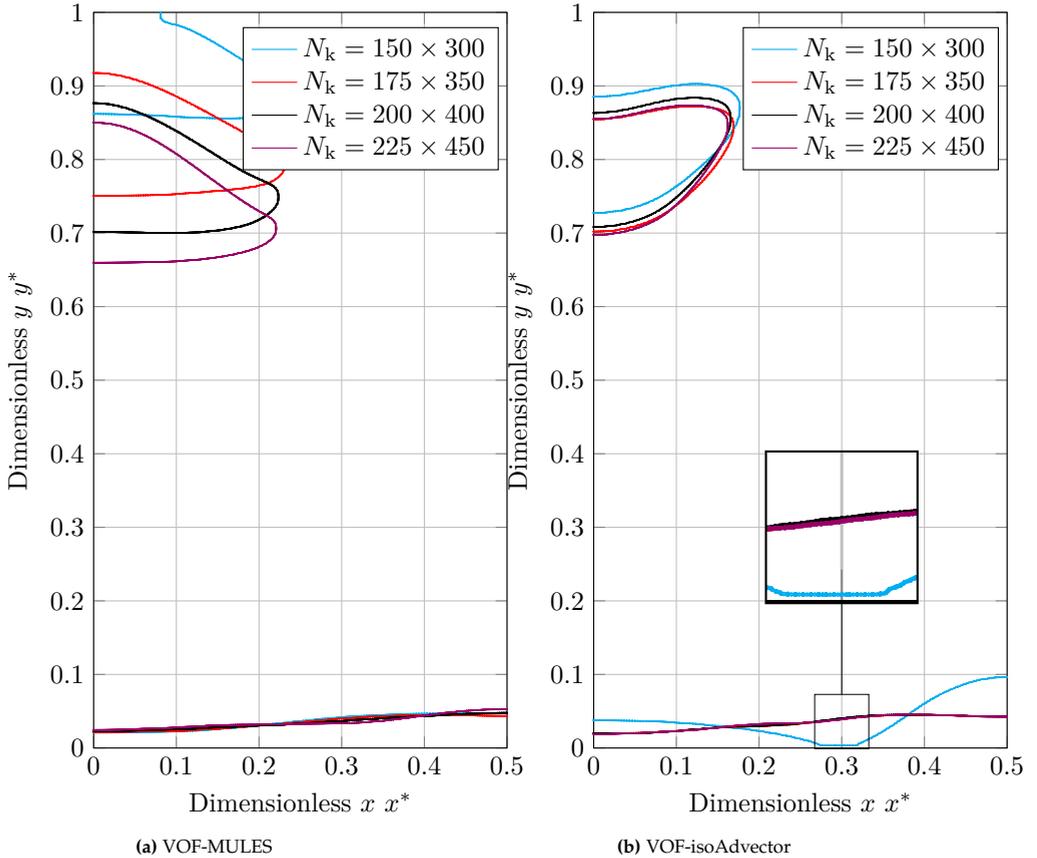
The primary motivation for including this particular benchmark case in the study is its utility in evaluating the performance of different interface description methods, especially in challenging scenarios. Specifically, this case provides insights into how these methods behave when the fluid-fluid interface is parallel and proximate to a wall.

Fig. 2.16 compares the shape of the first detached bubble at a specific time using VOF-MULES and VOF-isoAdvecter. Although both algorithms predict similar bubble sizes, they yield considerably different bubble shapes and bottom curvatures. The separation and rising of the bubbles cause an upward flow and a low-pressure area, which in turn change the bottom curvature of the bubbles and produce vortices at their sharp edges. Both the VOF-MULES and VOF-isoAdvecter methods generate upward flows, but their magnitudes and characteristics differ due to the presence of parasitic currents. This explains the distinct bubble shapes produced by the VOF-MULES and VOF-isoAdvecter methods.

The film predicted using VOF-isoAdvecter at the coarsest grid ( $N_k = 150 \times 300$ ) is another point of interest, as shown in Fig. 2.16b. The interface in this graph is connected to the bottom wall, as illustrated in the magnified image. This behavior (attached interface to the wall) is not observed with the VOF-MULES and various grid densities (Fig. 2.16a). When the grid structure is too coarse and the interface is too close to the wall, it is evident that the VOF-isoAdvecter technique does not accurately predict the interface behavior. Consequently, the VOF-isoAdvecter results for the  $N_k = 150 \times 300$  grid structure will be excluded from the subsequent Nusselt number results.

## 2.5. Benchmark cases

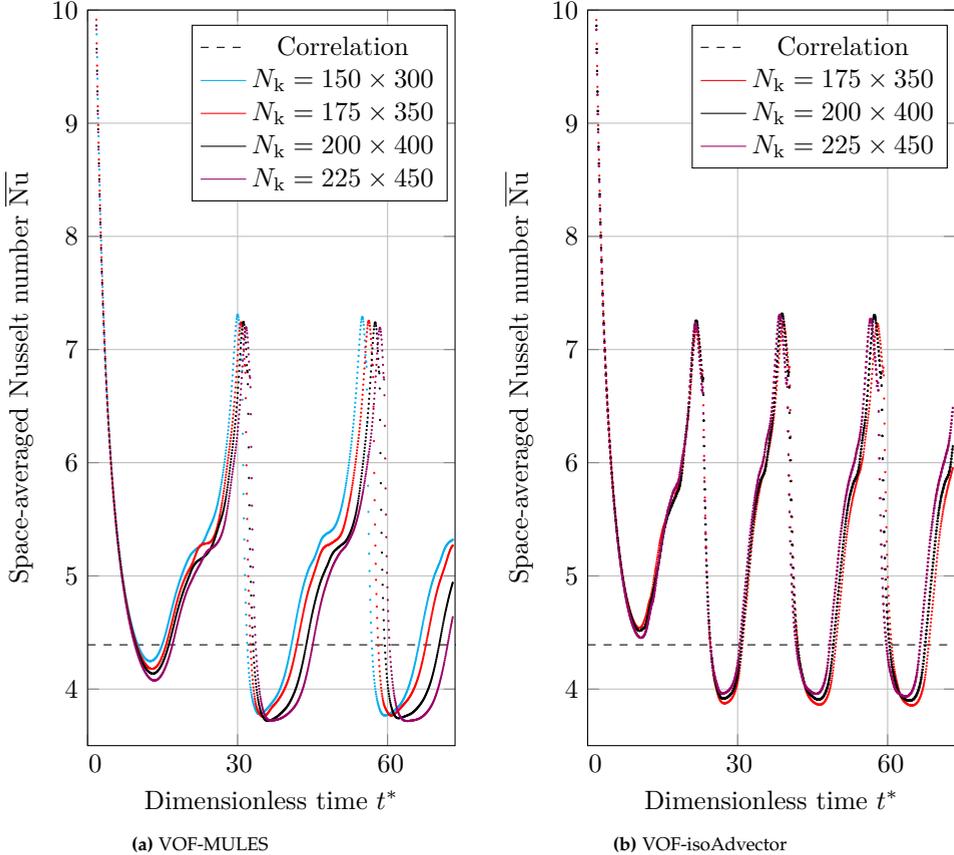
From Fig. 2.16b, for the VOF-isoAdvector, a grid structure of  $175 \times 350$  can be considered optimal (Grid size  $\approx 6.6 \times 6.6 \times 10^{-6} \text{ mm}^2$ ), while for VOF-MULES, the optimum mesh should be finer than  $225 \times 450$  for modeling the simulation (Grid size  $< 5 \times 5 \times 10^{-6} \text{ mm}^2$ ). It should also be mentioned that using extremely refined meshes, such as  $300 \times 600$ , results in the bubble not separating from the film. This indicates that further refining the mesh beyond certain limits does not yield accurate or desirable results in the simulation.



**Fig. 2.16:** A representation of vapor interface (film and bubble), obtained by VOF-MULES and VOF-isoAdvector for 2D film boiling problem at a specified height and different grid sizes ( $N_k = [150 \times 300, 175 \times 350, 200 \times 400, 225 \times 450]$ ). Adopted from Yahyaee et al. [30] © 2022 Licensee MDPI, Basel, Switzerland.

The space-averaged Nusselt value in dimensionless time is demonstrated in Fig. 2.17. The Nusselt number is very sensitive to film thickness. Heat flux is high when the vapour film is thin and low when the film is thick. The average flux flow and Nusselt number both rise as the vapour accelerates

to fill the bubble, thinning the layer that remains. The Nusselt number is reduced because the film thickness is increased when the vapour reattaches to the superheated wall following separation.



**Fig. 2.17:** The representation of space-averaged Nusselt number, obtained by numerical (VOF-MULES and VOF-isoAdvectord) and Berenson correlation (equation 2.38) for 2D film boiling problem and with different grid sizes. Adopted from Yahyae et al. [30] © 2022 Licensee MDPI, Basel, Switzerland.

## 2.5.5 Stationary 3D Spherical Bubble Scenario

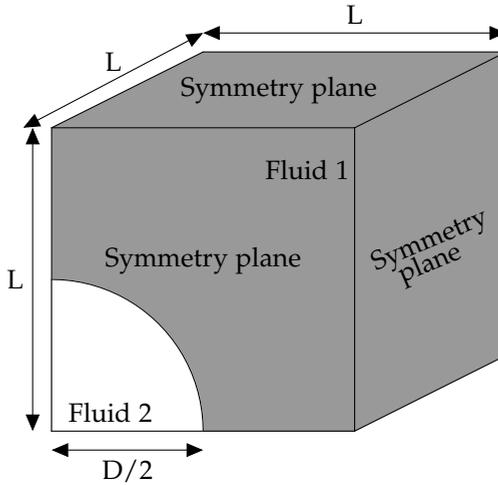
The test case under discussion explores a three-dimensional, gravity-neutral, non-evaporating bubble, illustrated in Fig. 2.18. The computational domain is cubic, filled with fluid type 1, and encloses a spherical bubble composed of fluid type 2. Computational efficiency is optimized by focusing on one-eighth of the full 3D sphere, confined within a cubic domain of dimensions  $L \times L \times L \text{ m}^3$ . For this case,  $L = 1 \text{ m}$ , and the spherical bubble has a diameter

## 2.5. Benchmark cases

$D$  equal to the cube's edge length  $L$ . Pertinent fluid properties are outlined in Table 2.4, and a grid of  $64 \times 64 \times 64$  structured cells is employed for the simulation.

The objective of incorporating this static bubble scenario is to facilitate the examination of non-physical currents in close proximity to the fluid-fluid interface. In contrast to cases involving thermal phase changes, where considerable interface motion obscures the evaluation of such spurious effects, this gravity-neutral setup serves as an ideal platform for an isolated examination.

Emphasis in the analysis is placed on peak velocity magnitudes and pressure distribution patterns within the computational domain.



**Fig. 2.18:** Geometry and boundary conditions of 3D static bubble case in zero gravity condition.

**Table 2.4:** Fluid properties used to solve 3D static bubble case benchmark case.

	Dimension	Liquid	Vapor
Density, $\rho$	$\text{kg m}^{-3}$	1	1
Viscosity, $\mu$	$\text{Pa s}$	$1 \times 10^{-2}$	$1 \times 10^{-2}$
Surface tension, $\sigma$	$\text{N m}^{-1}$	1	

The peak velocity is normalized by the capillary velocity ( $U_\sigma$ ), represented as the highest Capillary number ( $\text{Ca}_{\max}$ ) in Fig. 2.19. The time scale is non-dimensionalized using the capillary time ( $t_\sigma$ ), as previously defined in equation 2.23.

Despite the lack of external forces, Fig. 2.19 displays spurious non-physical velocities within the computational domain. This underscores the need for

careful interpretation and possible refinements in the numerical approach.

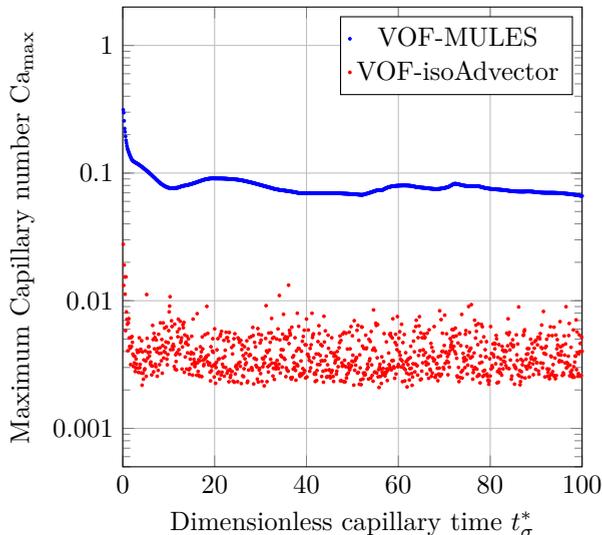


Fig. 2.19: Maximum capillary number for the 3D static bubble case.

Spurious non-physical velocities can emerge due to a numerical imbalance between the errors in approximating the pressure gradient and surface tension forces in the Navier-Stokes equations. This imbalance introduces an extraneous term in the vorticity equation, consequently generating non-physical velocities. Both VOF-MULES and VOF-isoAdvecter display oscillations in these spurious velocities, and their mean value remains relatively stable instead of diminishing to negligible levels. This implies that the ongoing imbalance between the pressure gradient and surface tension forces perpetuates these unwanted currents. Importantly, Fig. 2.19 indicates that VOF-isoAdvecter tends to produce non-physical velocities of reduced magnitude in comparison to VOF-MULES.

In examining this three-dimensional static bubble case, attention is drawn to the variation of pressure across the domain. This pressure variation is quantified through the dimensionless pressure coefficient ( $C_p$ ) and plotted in Fig. 2.20. Additionally, the  $x$  coordinate is non-dimensionalized using the bubble diameter  $D$ .

According to the Young-Laplace equation, the pressure differential across a curved surface due to surface tension is captured as

$$\Delta p = \sigma \left( \frac{1}{R_1} + \frac{1}{R_2} \right), \quad (2.39)$$

where  $R_1$  and  $R_2$  represent the radii in two orthogonal directions. In this study, for a spherical bubble, these radii are identical and equal to 0.5 m.

## 2.5. Benchmark cases

Consequently, the pressure differential  $\Delta p$  is 4 Pa, resulting in a dimensionless pressure coefficient of  $8 \times 10^{-4}$ .

Inspection of Fig. 2.20 reveals that the VOF-isoAdvector method offers superior accuracy in predicting the dimensionless pressure coefficient,  $\Delta p$ , and by extension, the curvature ( $\kappa$ ). Specifically, in the vicinity of the interface ( $x^* = 0.5$ ), VOF-isoAdvector more closely aligns with the theoretical pressure coefficient, unlike the VOF-MULES approach which exhibits a more oscillatory behavior around this point.

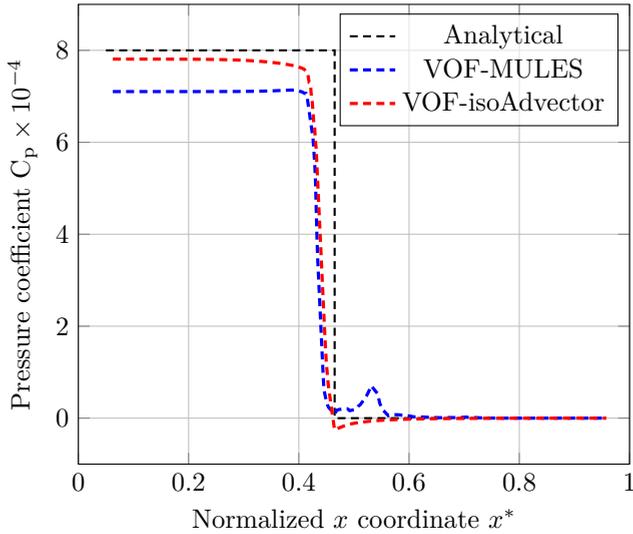
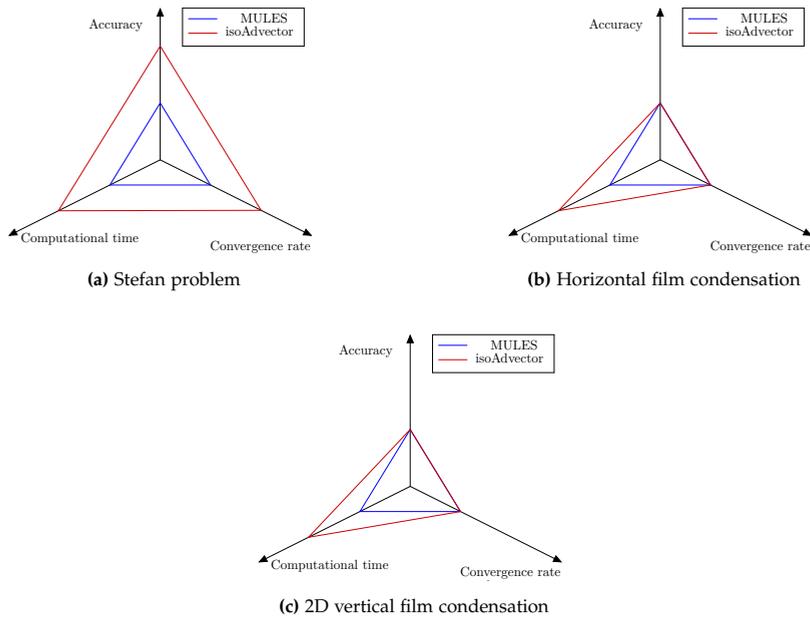


Fig. 2.20: Dimensionless pressure jump across the 3D static bubble.

## Chapter 2. isoAdvectord approach to describe the gas-liquid interface



**Fig. 2.21:** VOF-MULES and VOF-isoAdvectord algorithms' performances in three thermal phase change benchmark instances, shown graphically in terms of accuracy, computing time, and convergence rate. Each axis depicts a study domain and has a pair of values. Methods with better performance in a given region will be assigned a higher value, while those with less performance will be assigned a lower value. Adopted from Yahyae et al. [30] © 2022 Licensee MDPI, Basel, Switzerland.

## 2.6 Conclusion

This chapter has critically examined the interface capturing methods used in CFD for multiphase flows, specifically within the context of OpenFOAM's prevalent thermal phase change solvers. Traditionally, these solvers have employed the VOF-MULES method, which, despite its effectiveness in a wide range of applications, exhibits limitations such as smeared interfaces and the generation of spurious currents near the gas-liquid interface.

To overcome these challenges, this study has introduced and developed the 'thermalPhaseChangeFlow' solver<sup>2</sup>, which integrates the VOF-isoAdvectord method. The 'thermalPhaseChangeFlow' solver, evolving from the base of interPhaseChangeFoam, interFlow, and phaseChangeHeatFoam solvers, enhances the simulation accuracy, particularly in scenarios involving thermal phase changes.

<sup>2</sup>The solver is accessible at the following URL: <https://github.com/AAU-OpenFOAM/LSThermalPhaseChangeFlow>

## 2.6. Conclusion

A comparative analysis of 'thermalPhaseChangeFlow' and the twoPhaseFlow solver, which also utilizes isoAdvector for interface capturing for thermal phase change phenomena, was conducted. This comparison focused on a 2D bubble case with varying surface contact angles. It was observed that while twoPhaseFlow is effective in many situations, it encounters difficulties in accurately simulating varying contact angles. In contrast, 'thermalPhaseChangeFlow' demonstrates extended functionality and improved accuracy in these scenarios.

Comparative evaluations of the 'thermalPhaseChangeFlow' solver against the VOF-MULES method were performed through a series of thermal benchmark cases, such as the Stefan problem, horizontal film condensation, condensation on a vertical plate, and 2D film boiling. As documented in Section 2.3, these tests were conducted using uniform numerical schemes and solution algorithms. As shown in Fig. 2.21, the results indicated that the VOF-isoAdvector, integrated within 'thermalPhaseChangeFlow', achieved slightly superior or comparable accuracy and convergence rates in thermal phase change benchmarks while requiring less computational time than the traditional VOF-MULES approach.

However, it is important to note that in the 2D film boiling benchmark, VOF-isoAdvector encounters limitations, particularly in coarser grid setups and where the interface is closely aligned parallel to the wall.

An additional investigation was conducted using a non-thermal benchmark—the "Stationary 3D Spherical Bubble Scenario"—to assess the reduction of spurious currents near the interface and curvature prediction. The results from this scenario indicate that VOF-isoAdvector improves curvature prediction and minimizes spurious currents compared to VOF-MULES.

## References

- [1] C. Kunkelmann and P. Stephan, "CFD simulation of boiling flows using the volume-of-fluid method within OpenFOAM," *Numerical Heat Transfer, Part A: Applications*, vol. 56, no. 8, pp. 631–646, 2009.
- [2] S. Hardt and F. Wondra, "Evaporation model for interfacial flows based on a continuum-field representation of the source terms," *Journal of Computational Physics*, vol. 227, no. 11, pp. 5871–5895, 2008.
- [3] M. Nabil and A. S. Rattner, "A Computational Study on the Effects of Surface Tension and Prandtl Number on Laminar-Wavy Falling-Film Condensation," *Journal of Heat Transfer*, vol. 139, no. 12, pp. 1–11, 2017.
- [4] N. Samkhaniani and M. R. Ansari, "Numerical simulation of superheated vapor bubble rising in stagnant liquid," *Heat and Mass Transfer*, vol. 53, no. 9, pp. 2885–2899, 2017.
- [5] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti, "Modelling Merging and Fragmentation in Multiphase Flows with SURFER," *Journal of Computational Physics*, vol. 113, no. 1, pp. 134–147, 7 1994.
- [6] L. Gamet, M. Scala, J. Roenby, H. Scheufler, and J. L. Pierson, "Validation of volume-of-fluid OpenFOAM® isoAdvector solvers using single bubble benchmarks," *Computers and Fluids*, vol. 213, p. 104722, 12 2020. [Online]. Available: <https://doi.org/10.1016/j.compfluid.2020.104722>
- [7] J. Roenby, H. Bredmose, and H. Jasak, "IsoAdvector: Geometric VOF on general meshes," in *OpenFOAM®*. Springer, 2019, pp. 281–296.
- [8] K. Missios, N. G. Jacobsen, and J. Roenby, "Using the isoAdvector Geometric VOF Method for Interfacial Flows Through Porous Media Using the isoAdvector Geometric VOF Method for Interfacial Flows Through Porous Media," *9th Conference on Computational Methods in Marine Engineering (Marine 2021)*, no. June, 2021.
- [9] J. Roenby, H. Bredmose, and H. Jasak, "A computational method for sharp interface advection," *Royal Society Open Science*, vol. 3, no. 11, p. 160405, 2016.
- [10] V. Vukčević, J. Roenby, I. Gatin, and H. Jasak, "A sharp free surface finite volume method applied to gravity wave flows," *arXiv*, vol. M, 2018.
- [11] H. Scheufler and J. Roenby, "TwoPhaseFlow: An OpenFOAM based framework for development of two phase flow solvers," 2021. [Online]. Available: <https://github.com/DLR-RY/TwoPhaseFlow>
- [12] J. Roenby, B. Eltard Larsen, H. Bredmose, and H. Jasak, "IS-New CFD methods for marine hydrodynamics A new Volume-of-Fluid method in OpenFOAM A NEW VOLUME-OF-FLUID METHOD IN OPENFOAM," Tech. Rep., 2017.
- [13] K. Missios, N. Jacobsen, K. Moeller, and J. Roenby, "Extending the isoAdvector Geometric VOF Method to Flows in Porous Media," *OpenFOAM® Journal*, vol. 3, pp. 66–74, 5 2023.

## References

- [14] H. G. Weller, "A new approach to VOF-based interface capturing methods for incompressible and compressible flow," *OpenCFD Ltd., Report TR/HGW*, vol. 4, p. 35, 2008.
- [15] N. Samkhaniani and M. R. Ansari, "The evaluation of the diffuse interface method for phase change simulations using OpenFOAM," *Heat Transfer - Asian Research*, vol. 46, no. 8, pp. 1173–1203, 2017.
- [16] J. U. Brackbill, D. B. Kothe, and C. Zemach, "A continuum method for modeling surface tension," *Journal of computational physics*, vol. 100, no. 2, pp. 335–354, 1992.
- [17] M. Dianat, M. Skarysz, and A. Garmory, "A Coupled Level Set and Volume of Fluid method for automotive exterior water management applications," *International Journal of Multiphase Flow*, vol. 91, pp. 19–38, 5 2017.
- [18] J. Klostermann, K. Schaake, and R. Schwarze, "Numerical simulation of a single rising bubble by VOF with surface compression," *International Journal for Numerical Methods in Fluids*, vol. 71, no. 8, pp. 960–982, 2013.
- [19] A. Albadawi, D. B. Donoghue, A. J. Robinson, D. B. Murray, and Y. M. Delauré, "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*, vol. 53, pp. 11–28, 7 2013.
- [20] M. Aboukhedr, A. Georgoulas, M. Marengo, M. Gavaises, and K. Vogiatzaki, "Simulation of micro-flow dynamics at low capillary numbers using adaptive interface compression," *Computers & Fluids*, vol. 165, pp. 13–32, 2018.
- [21] E. Olsson, "A description of isoAdvector-a numerical method for improved surface sharpness in two-phase flows," *Proceedings of CFD with OpenSource Software, Edited by Nilsson. H*, 2017.
- [22] I. Tanasawa, "Advances in condensation heat transfer," in *Advances in heat transfer*. Elsevier, 1991, vol. 21, pp. 55–139.
- [23] M. Knudsen and J. R. Partington, "The Kinetic. Theory of Gases. Some Modern Aspects," *The Journal of Physical Chemistry*, vol. 39, no. 2, p. 307, 2002.
- [24] W. H. Lee, "Pressure iteration scheme for two-phase flow modeling," IN "MULTIPHASE TRANSPORT: FUNDAMENTALS, REACTOR SAFETY, APPLICATIONS"., pp. 407–432, 1980.
- [25] A. Yahyaee, A. S. Bahman, K. Olesen, and H. Sørensen, "Level-Set Interface Description Approach for Thermal Phase Change of Nanofluids," *Nanomaterials*, vol. 12, no. 13, p. 2228, 2022.
- [26] N. Samkhaniani and M. R. Ansari, "Numerical simulation of bubble condensation using CF-VOF," *Progress in Nuclear Energy*, vol. 89, pp. 120–131, 2016. [Online]. Available: <http://dx.doi.org/10.1016/j.pnucene.2016.02.004>
- [27] D. Z. Guo, D. L. Sun, Z. Y. Li, and W. Q. Tao, "Phase change heat transfer simulation for boiling bubbles arising from a vapor film by the VOSET method," *Numerical Heat Transfer; Part A: Applications*, vol. 59, no. 11, pp. 857–881, 2011.
- [28] E. O. Fogliatto, A. Clause, and F. E. Teruel, "Development of a double-MRT pseudopotential model for tridimensional boiling simulation," *International Journal of Thermal Sciences*, vol. 179, p. 107637, 9 2022.

## References

- [29] Z. Pan, J. A. Weibel, and S. V. Garimella, "A saturated-interface-volume phase change model for simulating flow boiling," *International Journal of Heat and Mass Transfer*, vol. 93, pp. 945–956, 2 2016.
- [30] A. Yahyaee, A. S. Bahman, and H. Sørensen, "A Benchmark Evaluation of the isoAdvection Interface Description Method for Thermally–Driven Phase Change Simulation," *Nanomaterials*, vol. 12, no. 10, p. 1665, 2022.
- [31] A. S. Rattner and S. Garimella, "Simple mechanistically consistent formulation for volume-of-fluid based computations of condensing flows," *Journal of Heat Transfer*, vol. 136, no. 7, pp. 1–9, 2014.
- [32] W. Nusselt, "Die oberflächenkondensation des wasserdampfes," *VDI-Zs*, vol. 60, p. 541, 1916.
- [33] D. Juric and G. Tryggvason, "Computations of boiling flows," *International journal of multiphase flow*, vol. 24, no. 3, pp. 387–410, 1998.
- [34] F. Gibou, L. Chen, D. Nguyen, and S. Banerjee, "A level set based sharp interface method for the multiphase incompressible Navier–Stokes equations with phase change," *Journal of Computational Physics*, vol. 222, no. 2, pp. 536–555, 3 2007.
- [35] A. Begmohammadi, M. H. Rahimian, M. Farhadzadeh, and M. A. Hatani, "Numerical simulation of single- and multi-mode film boiling using lattice Boltzmann method," *Computers & Mathematics with Applications*, vol. 71, no. 9, pp. 1861–1874, 5 2016.
- [36] S. Gong and P. Cheng, "Direct numerical simulations of pool boiling curves including heater's thermal responses and the effect of vapor phase's thermal conductivity," *International Communications in Heat and Mass Transfer*, vol. 87, pp. 61–71, 10 2017.
- [37] A. K. Dixit, R. Kumar, and A. K. Das, "Investigation of film boiling at a liquid-liquid contact," *International Journal of Heat and Mass Transfer*, vol. 196, p. 123234, 11 2022.
- [38] P. J. Berenson, "Film-boiling heat transfer from a horizontal surface," *J Heat Transf*, vol. 83, no. 3, p. 351–356, 1961.
- [39] A. Esmaeeli and G. Tryggvason, "Computations of film boiling. Part I: numerical method," *International journal of heat and mass transfer*, vol. 47, no. 25, pp. 5451–5461, 2004.

## Chapter 3

# CLSVOF method to describe the gas-liquid interface

Identifying the gas-liquid interface accurately is a critical challenge in modeling thermal phase change flows. While the VOF approach is commonly used for such simulations, it can yield erroneous curvature computations. This chapter explores an alternative: the coupled level-set VOF (CLSVOF) method, previously introduced by Bourlioux [1]. CLSVOF combines the mass-conserving attributes of VOF with the level-set method's accuracy in curvature computation at the interface.

The primary aim of this chapter is to assess the effectiveness of the CLSVOF method in simulating thermal phase change phenomena, through comparisons with the traditional VOF method across established benchmark cases. While CLSVOF has received considerable attention in previous studies, most research has been conducted using proprietary, in-house developed solvers that are not generally accessible to the public. Furthermore, when these solvers are available, they are typically designed for two-phase flow scenarios and do not possess the functionality for simulating thermal phase changes. Consequently, for the objectives of this study, it was imperative to develop a new thermal phase change solver that incorporates CLSVOF for interface capturing, signifying an initial yet crucial phase of this study.

Benchmark scenarios for this study include those in Section 2.5. The chapter concludes by contrasting the results from the CLSVOF and VOF methods, highlighting their respective strengths and weaknesses in different thermal phase change contexts.

This chapter is organized as follows: Section 3.1 provides a literature review on the level-set method. Section 3.2 presents the governing equations for the CLSVOF method. Section 3.3 discusses the benchmark cases, including the Stefan problem (3.3.1), horizontal film condensation problem (3.3.2),

film condensation on a vertical plate (3.3.3), 2D film boiling (3.3.4), and Stationary 3D Spherical Bubble Scenario (3.3.5). Finally, Section 3.4 concludes the chapter, summarizing the findings and implications of the study.

### 3.1 Literature Review On Level Set And CLSVOF Methods

Originally conceptualized by Osher and Sethian [2], the level-set method found its early applications in multiphase incompressible flows through the work of Sussman et al. [3]. This method has since evolved, as demonstrated by Losasso et al. [4], into an effective tool for simulating multiphase flows with high precision. The level-set method operates differently from traditional methods that use a continuous volume fraction variable. It employs a signed distance function, denoted as  $\psi$ , which helps to distinctly categorize the fluids within the mixture. In this function, positive values represent one fluid, negative values the other, and a zero value indicates the interface's location. The advection of this interface is governed by solving a transport equation for  $\psi$ , which is periodically reset to maintain its characteristic as a distance function [5]. The distinct advantage of the level-set method lies in its ability to sharply define the interface and precisely calculate interfacial properties like normal vectors and curvature, critical for accurately modeling surface tension forces. Nevertheless, unlike the VOF methods, the level-set approach does not inherently conserve mass. Studies by Sussman et al. [3] have pointed out mass loss issues within the level-set method, attributed to the reinitialization process that may inadvertently shift the interface [6].

In both VOF and level-set methods, accurately rendering surface tension forces at the interface between phases presents a significant numerical challenge. Typically, the surface tension is incorporated as a force term in the momentum equation, following the Continuous Surface Force (CSF) approach developed by Brackbill et al. [7]. This approach necessitates estimating the interface curvature and the normal vector at the interface, derived from the gradient of either the VOF or level-set function. Given the discontinuous nature of the interface, obtaining precise derivative calculations is a complex task. In VOF methods, where diffusion can blur the interface, accurately capturing the interface curvature is especially challenging. Conversely, level-set methods, while offering a distinct interface, concentrate the volumetric surface force in a narrow band around the interface, which can lead to numerical instabilities in calculating the normal vector. These computational issues often result in artificial vortical flows at the interface, known as spurious currents [8–10]. To mitigate these spurious currents, several strategies have been proposed, including enhancing curvature computation, ensuring dis-

### 3.2. CLSVOF governing equations

crete balance between the surface tension and pressure gradient forces, and employing adaptive time integration schemes to manage the high stiffness caused by surface tension. One effective approach is based on minimal energy principles, which can eliminate spurious currents to the precision level of the machine's calculation capacity. Another suggestion involves using consistent volume fluxes within the Navier-Stokes equations, applicable in both VOF [11] and level-set frameworks [12], grounded on geometric reasoning.

As the field of numerical methods progresses, the precision of various techniques has been notably improved. These advancements primarily concentrate on the periodic reinitialization of the level-set function, a process crucial for resolving the challenges related to mass conservation. Key among these strategies are the Height Function (HF) method and the conservative level-set approach, as explored in works by Francois et al. [8, 13] and Olsson et al. [14]. These methods have been successful in achieving a second-order accurate calculation of curvature, significantly reducing the issues associated with mass conservation in multiphase flow simulations.

Hybrid modeling approaches, combining the features of the level-set and VOF methods, have been developed to tackle the issue of mass conservation in level-set simulations. These hybrid models capitalize on the mass-conserving aspect of the VOF method, particularly beneficial for simulations on coarse grids, while integrating the level-set method's finesse in defining the interface. The level-set's capability to offer a smooth, differentiable domain is advantageous for calculating surface tension forces effectively. The Combined Level Set and Volume of Fluid (CLSVOF) approach, which amalgamates the advantages of both methods, was initially proposed by Bourlioux [1] and later refined and popularized by Sussman and Puckett [15].

The CLSVOF technique has seen diverse implementations across various research domains. Sussman and Puckett's fully coupled CLSVOF method [15] set the precedent, leading to its widespread application in numerous studies [16, 17]. Its effectiveness has been validated on cartesian orthogonal meshes, with notable examples being Ménard et al.'s [18] work on diesel jet atomization and Yokoi's [19] research on droplet splashing and Rayleigh–Taylor instability. Furthermore, Arienti and Sussman [20] have extended the CLSVOF method to adaptive cartesian meshes. Recent advancements have also seen the application of CLSVOF on unstructured meshes, broadening its scope and utility in complex flow simulations [21, 22].

## 3.2 CLSVOF Governing Equations

The governing equations for the flow, in the context of the CLSVOF method, build upon those introduced in section 2.3. These consist of the conservation equations for mass, momentum, energy, the advection equation for interface

description, and the governing equations for the VOF method. This section focuses on governing equations with particular emphasis on the level-set technique.

In the CLSVOF method, two scalar fields are defined: the level-set function ( $\psi$ ) and the fluid volume fraction ( $\alpha$ ). The interface location is defined by the set of points where  $\psi = 0$  (see Fig. 3.1).

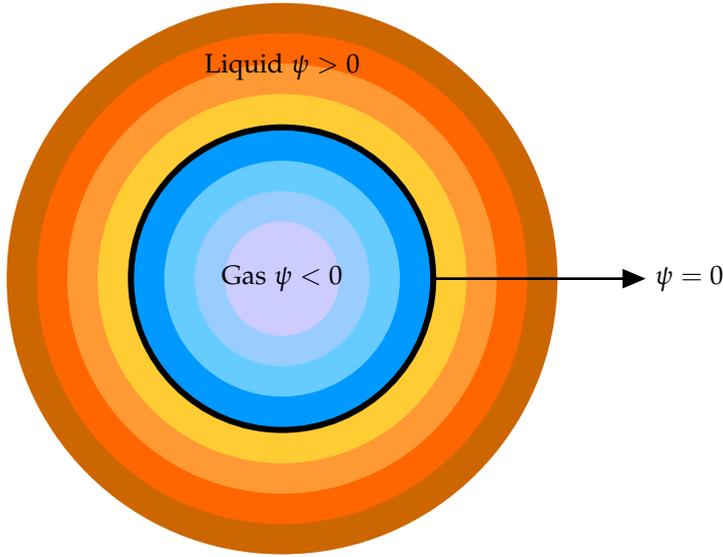


Fig. 3.1: A sketch of a level-set field, adopted from Omar et al. [23]

The initial  $\psi$  value is determined using the  $\alpha$  field under the assumption that the interface is located at the iso-line  $\alpha = 0.5$ . This is given by the equation:

$$\psi_0 = (2\alpha - 1)\Gamma, \quad (3.1)$$

where  $\Gamma$  is a small non-dimensional number defined by:

$$\Gamma = 0.75\Delta x, \quad (3.2)$$

and  $\Delta x$  is the mesh step size. The reason behind choosing equation 3.2 as the value of  $\Gamma$  is to set an initial  $\psi$  value close to the mesh step size. As seen in Fig. 3.1, this initial function is a signed function that separates the vapour and liquid phases at the points where  $\psi < 0$  and  $\psi > 0$ , respectively.

For the preservation of stable numerical outcomes and a uniform interface width, maintaining the level-set function to closely mimic a signed distance

### 3.2. CLSVOF governing equations

function — denoted by  $|\nabla\psi| = 1$  — proves to be of utmost importance. This condition can be achieved by subjecting the initial value of  $\psi$  to a repetitive process of re-initialization using the associated equation:

$$\frac{\partial\psi}{\partial\tau} - \text{sgn}(\psi_0)(1 - |\nabla\psi|) = 0, \quad (3.3)$$

where  $\text{sgn}(\psi_0)$  is a signed function, and  $\tau$  is artificial time. These are defined as:

$$\text{sgn}(\psi_0) = \frac{\psi_0}{|\psi_0|}, \quad (3.4)$$

and

$$\tau = 0.1\Delta x. \quad (3.5)$$

A more stable definition for  $\text{sgn}(\psi_0)$  is:

$$\text{sgn}(\psi_0) = \frac{\psi_0}{\sqrt{\psi_0^2 + \epsilon}}, \quad (3.6)$$

where

$$\epsilon = \max(\Delta x^2, \Delta y^2). \quad (3.7)$$

The re-initialization equation (equation 3.3) is solved  $\psi_{\text{corr}}$  times, where the number of iterations ( $\psi_{\text{corr}}$ ) is calculated as:

$$\psi_{\text{corr}} = \frac{\epsilon}{\Delta\tau}, \quad (3.8)$$

In this equation,  $\epsilon$  represents the interface thickness and is defined as:

$$\epsilon = 1.5\Delta x. \quad (3.9)$$

The equation for determining surface tension is given as:

$$\mathbf{F}_\sigma = \sigma\kappa(\psi)\delta(\psi)\nabla\psi. \quad (3.10)$$

In this equation, the symbol  $\sigma$  stands for the coefficient associated with surface tension,  $\kappa(\psi)$  signifies the curvature, and  $\delta(\psi)$  symbolizes the Dirac function. The function of Dirac, which restricts the effect of surface tension within the interface and stands as zero in both fluids, is represented as:

$$\delta(\psi) = \begin{cases} \frac{1}{2\epsilon} \left( 1 + \cos\left(\frac{\pi\psi}{\epsilon}\right) \right) & \text{if } |\psi| \leq \epsilon \\ 0 & \text{if } |\psi| > \epsilon \end{cases}. \quad (3.11)$$

Geometric properties of the interface, such as the normal vector ( $\mathbf{N}$ ), and the curvature  $\kappa(\psi)$ , can be efficiently and accurately computed using a standard second-order central finite difference scheme. These are given by:

$$\mathbf{N} = \frac{\nabla\psi}{|\nabla\psi|}, \quad (3.12)$$

and

$$\kappa(\psi) = -\nabla \cdot \mathbf{N}. \quad (3.13)$$

The foundational equations and methodologies for the CLSVOF method, as discussed above, are based on the research of Sussman and Puckett [15], and Omar et al. [23]. For those seeking a thorough understanding of CLSVOF's implementation and theoretical framework, the works of Sussman and Puckett [15] and Omar et al. [23] are invaluable resources.

The solver central to this chapter has been developed using OpenFOAM version 2006. Its design is primarily based on the `interPhaseChangeFoam` solver from OpenFOAM, integrating aspects of the `phaseChangeHeatFoam` solver [24]—originally developed for OpenFOAM 2.2—and features from `sclsVOFFoam` [25], initially designed for OpenFOAM 2.3. For researchers and practitioners interested in replicating this study or exploring the solver in greater depth, both the solver and the reference results for the benchmark cases are made publicly available at <https://github.com/AAU-OpenFOAM/LSThermalPhaseChangeFlow>.

### 3.3 Benchmark Cases

This chapter extends the work presented in Chapter 2, specifically in sections 2.5. Given that the benchmark scenarios, discretization techniques, and solver configurations are consistent with those outlined in Chapter 2, this chapter will not reiterate those details. Instead, the emphasis here is on the comparative analysis of results obtained using the CLSVOF method in thermal phase change phenomena.

For the sake of continuity and to avoid redundancy, it is important to note that discretization methods and solver settings remain the same as those detailed in Chapter 2, section 2.3.3. The `'fvSolution'` and `'controlDict'` files used in this chapter are identical to those in Chapter 2 and can be found in the respective appendices mentioned there. The dimensionless numbers employed in this chapter are the same as those introduced in section 2.3.4 of Chapter 2. These dimensionless quantities continue to serve as the basis for comparing and scaling the simulation results, facilitating their broader applicability.

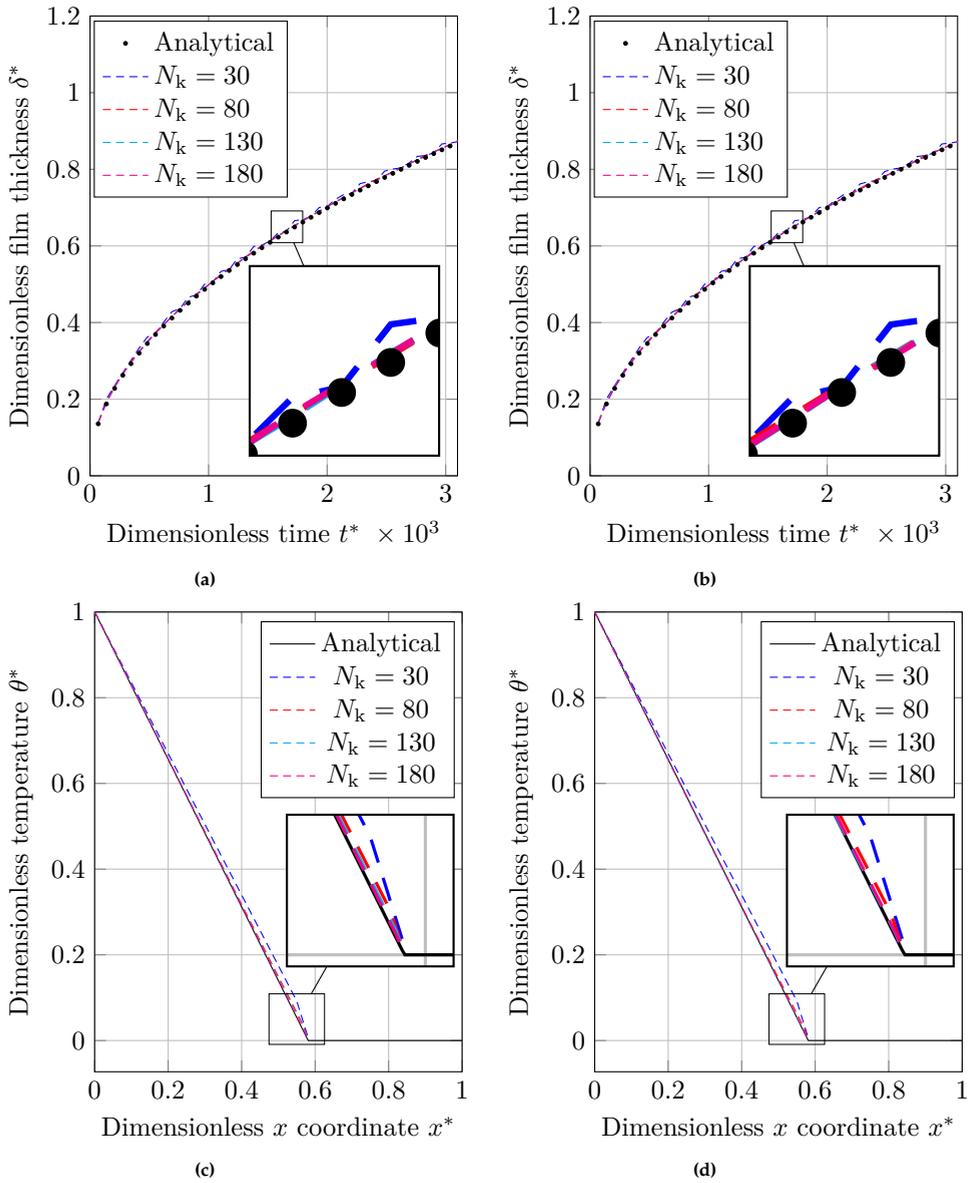
### 3.3.1 Stefan Problem

The Stefan problem benchmark case is addressed using  $N_k = [30, 80, 130, 180]$  grid counts, and the outcomes for both interface description methods, VOF and CLSVOF, are displayed in Fig. 3.2.

As shown in Fig. 3.2a and Fig. 3.2b, the dimensionless vapour film thickness ( $\delta^*$ ) graph follows a zigzag pattern when the coarsest grid structure ( $N_k = 30$ ) is used. Finer grid patterns smooth down this zigzag route (noises are dumped), and the resulting curves approach those of an analytical solution.

The dimensionless temperature across the domain at  $t = 20$  s can be seen in Fig. 3.2c and 3.2d. Coarser grid configurations ( $N_k = [30, 80]$ ) give dimensionless temperature that are larger than the analytical solution at areas close to the gas-liquid interface. This is because, as mentioned in previous studies [26, 27], the interface temperature is indeed set to saturation implicitly by the source term.

Examining these graphs for this particular Stefan problem benchmark, it becomes clear that under the given conditions and for a 1D domain length of  $L = 2$  mm, a mesh size corresponding to  $N_k = 130$ , which is  $1.54 \times 10^{-2}$  mm, emerges as the most suitable for simulations employing VOF and CLSVOF techniques between the mentioned grids structures. This selection is justified by the more stable and consistent results observed with  $N_k = 130$ . Such results show minimal fluctuation and are in close agreement with those obtained using a finer mesh of  $N_k = 180$ . Despite the increased refinement offered by  $N_k = 180$ , the improvements in simulation results are relatively incremental, confirming that the mesh with  $N_k = 130$  is a prudent choice for effective and precise simulations.



**Fig. 3.2:** A comparison between the numerical results obtained using the VOF and CLSVOF and the analytical solution for the Stefan problem. The comparisons include: Dimensionless thickness of the vapour film ( $\delta^*$ ) vs dimensionless time ( $t^*$ ), measured by (a) VOF and (b) CLSVOF; dimensionless temperature distribution ( $\theta^*$ ) across the dimensionless length ( $x^*$ ), obtained using (c) VOF and (d) CLSVOF.

Fig. 3.3 illustrates the L2 error concerning the temperature profile, providing a comparison of the efficacy of different methods. It is noticeable that

### 3.3. Benchmark Cases

both VOF and CLSVOF show a pattern of convergence when a finer grid structure is employed. In this context, CLSVOF exhibits a marginally lower error output than VOF, which implies its superior performance.

Taking into account the results plotted in Fig. 3.3, the rate of convergence can be evaluated using equation 2.29. In the comparative evaluation of VOF and CLSVOF methodologies across the refined grids ( $N_k = [130, 180]$ ), the analysis indicated a noticeable improvement in the convergence rate for the CLSVOF method compared to VOF. This finding suggests that CLSVOF exhibits a more efficient convergence behavior within this specific grid range, highlighting its potential for more accurate simulations in thermal phase change scenarios.

Moreover, Fig. 3.3 displays the computation time, represented via dashed lines, corresponding to each grid size. It is clear that, across all grid sizes, the VOF method outperforms in terms of computational speed.

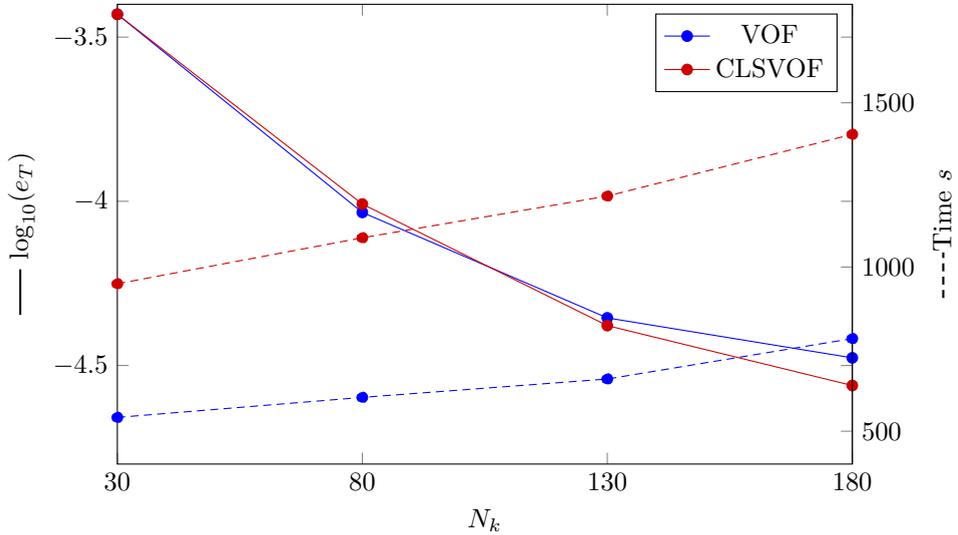


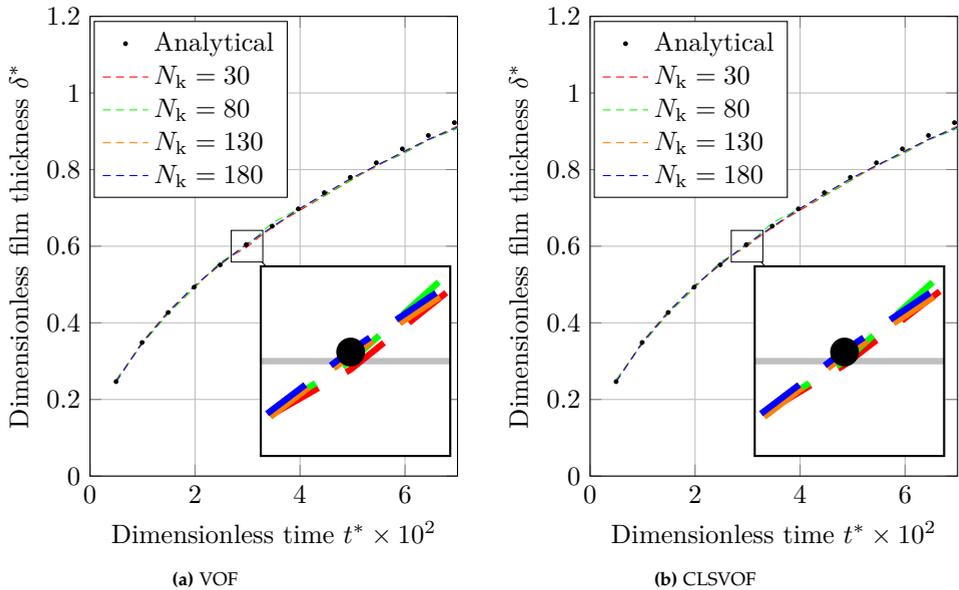
Fig. 3.3: Solid lines show logarithmic representation of the error  $e_T$  (equation 2.28) and Dashed lines show computation time of VOF and CLSVOF at various grid structures for solving the Stefan problem. The horizontal axis ( $N_k$ ) shows number of grids along the  $x$  axis. Adopted from Yahyaee et al. [28] © 2022 Licensee MDPI, Basel, Switzerland.

### 3.3.2 Horizontal Film Condensation Problem

This benchmark case is studied using four grid configurations with varied  $x$  grid counts ( $N_k = [30, 80, 130, 180]$ ). Fig. 3.4 shows the dimensionless evolution of the condensed liquid film thickness ( $\delta^*$ ) as a function of dimensionless time ( $t^*$ ), acquired by the VOF and CLSVOF techniques. Results are shown

to be in agreement with analytical results for all grid sizes with the exception of the coarsest one, which is associated with small deviations from the analytical result.

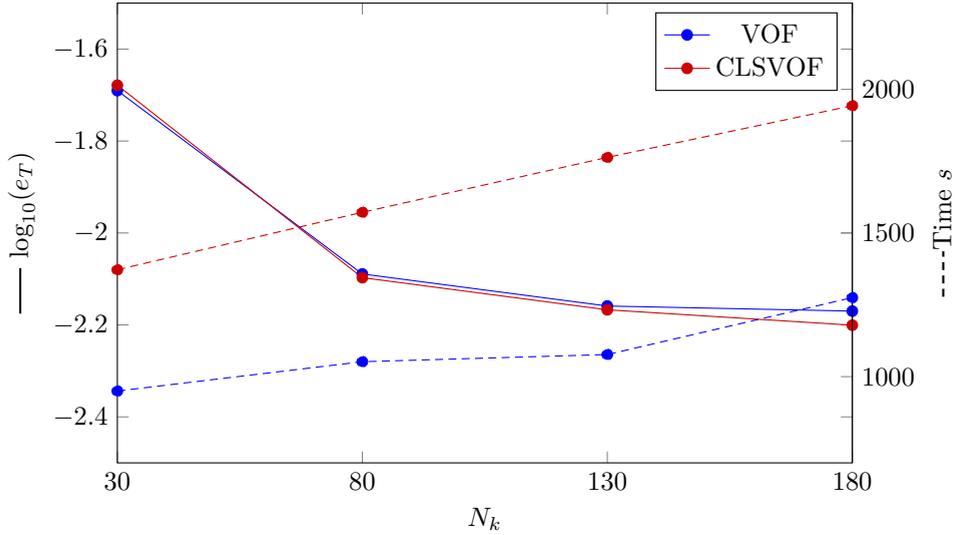
A look at the graphs for the horizontal film condensation benchmark suggests that a mesh size around  $2.3 \times 10^{-3}$  mm works well for simulations with a domain length of  $L = 0.3$  mm, which corresponds to a grid of  $N_k = 130$ . When we apply VOF and CLSVOF techniques with this mesh size, the simulations are stable and the results are smooth. This mesh,  $N_k = 130$ , outperforms larger mesh sizes and is almost as good as the finer  $N_k = 180$  mesh, but without the extra computational cost. This makes  $N_k = 130$  a smart choice for efficient and precise simulations in this study.



**Fig. 3.4:** Dimensionless thickness ( $\delta^*$ ) of the condensed liquid film with dimensionless time ( $t^*$ ) obtained by (a) VOF and (b) CLSVOF in the horizontal film condensation case.

As seen in Fig. 3.5, with grid configurations of  $N_k = [30, 80, 130]$ , both techniques provide the same level of precision. At the finest grid structure ( $N_k = 180$ ), VOF cannot match the CLSVOF and generates a greater error rate. In assessing the convergence rates (derived from equation 2.29) across the refined grid structures ( $N_k = [130, 180]$ ), a marked differentiation between the VOF and CLSVOF methods is observed. Notably, the CLSVOF method demonstrates an enhanced rate of convergence, outperforming the VOF method. Moreover, dashed lines represent the computing time analysis in Fig. 3.5. These results demonstrate that VOF provides quicker calculation than CLSVOF.

### 3.3. Benchmark Cases



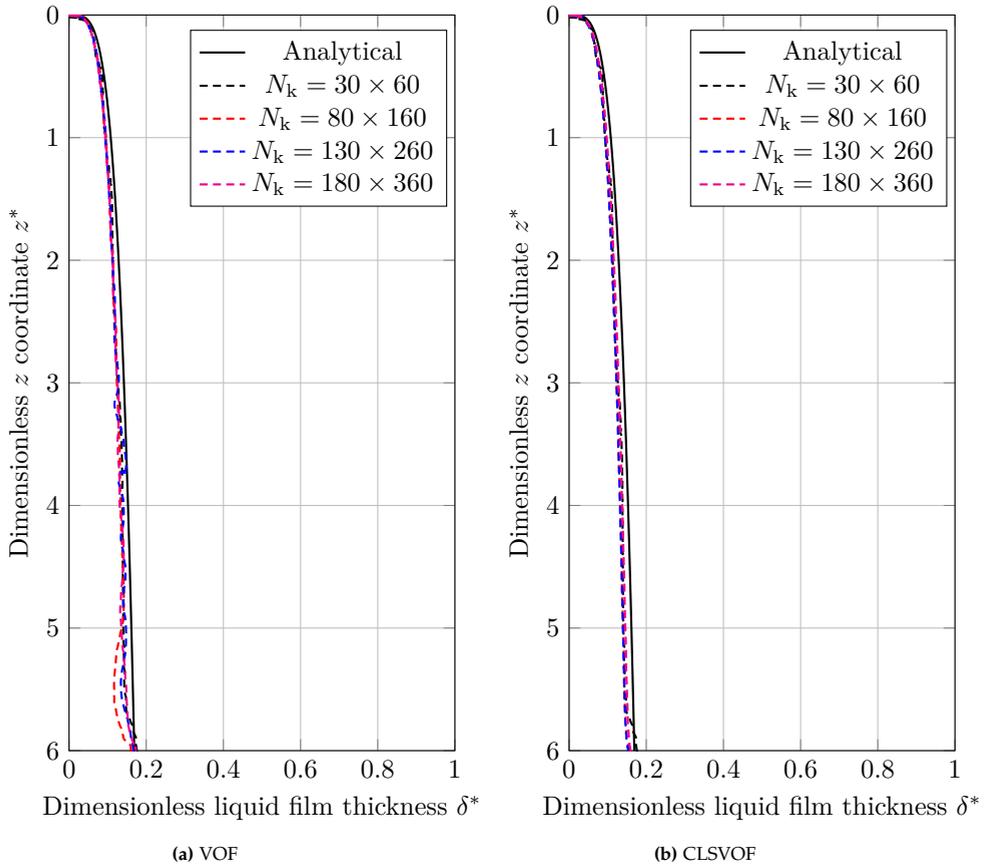
**Fig. 3.5:** Solid lines show the logarithmic representation of the error  $e_\delta$  (equation 2.31) and dashed lines reflect computation time of VOF and CLSVOF at various grid structures for solving horizontal film condensation benchmark case. The horizontal axis ( $N_k$ ) shows the number of grids along the  $x$  axis. Adopted from Yahyaee et al. [28] © 2022 Licensee MDPI, Basel, Switzerland.

#### 3.3.3 Laminar Film Condensation On A Vertical Plate

For this benchmark scenario, four different mesh configurations with  $N_k = [30 \times 60, 80 \times 160, 130 \times 260, 180 \times 360]$  are employed. The VOF and CLSVOF methods' results for the dimensionless condensed film thickness are shown in Fig. 3.6.

The simulation results of the CLSVOF method closely resemble the analytical ones, allowing for a precise prediction of the condensed film thickness. On the other hand, there is a noticeable discrepancy between the VOF technique's simulation results and the analytical data.

As the grid resolution increases, the accuracy of the simulation results improves, and they approach the analytical solution more closely.



**Fig. 3.6:** A representation of the condensed liquid film interface, obtained by numerical (VOF and CLSVOF) and analytical methods for laminar condensation on a vertical plate benchmark case and with different grid structures ( $N_k = [30 \times 60, 80 \times 160, 130 \times 260, 180 \times 360]$ ). The axes have been made dimensionless by scaling the values with respect to the length  $L$ .

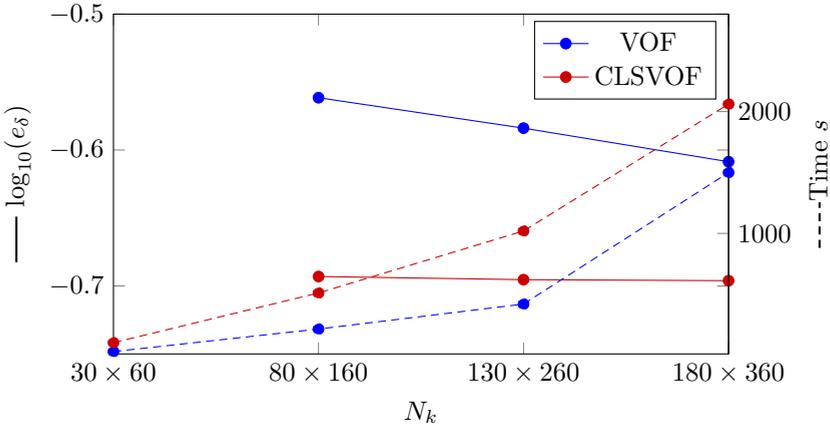
As demonstrated in Fig. 3.6a, a grid structure of  $130 \times 260$  provides an adequate balance between accuracy and computational efficiency for VOF simulations. Conversely, for CLSVOF simulations, Fig. 3.6b reveals that a grid configuration of  $80 \times 160$  achieves a similar equilibrium. As previously discussed in the last chapter, a non-uniform grid in OpenFOAM with simpleGrading parameters set to  $(4, 0.25, 1)$ , is used for the simulation of this benchmark case.

Fig. 3.7 presents the L2 error computation, offering a comparative perspective on the performance of VOF and CLSVOF methods. The grid configuration of  $30 \times 60$  is excluded from the computation of  $e_\delta$  and is not presented in Fig. 2.14. This decision stems from two observations at this grid size: the interface experiences substantial fluctuations, leading to a deviation from a

### 3.3. Benchmark Cases

stable linear profile; separately, the interface becomes overly thick and dispersed. In examining the performance across the finer grid configurations ( $N_k = [130 \times 260, 180 \times 360]$ ), the data reveals that while CLSVOF consistently demonstrates lower errors, VOF exhibits a more notable improvement in convergence rates. This observed pattern is logical; as the error magnitude for CLSVOF is already low, further reductions become increasingly challenging, which naturally results in a less dramatic convergence rate. This does not diminish the value of CLSVOF's precision but rather emphasizes the diminishing returns of error reduction in highly accurate models. The concept of "diminishing returns" in this context means that when a numerical model like CLSVOF already has a very low error, making further improvements to reduce the error becomes more difficult. Each additional effort to decrease the error yields smaller benefits. In other words, as the accuracy of the model increases, the amount of error reduction achieved with each refinement in the model (like using a finer grid) becomes smaller.

The analysis of computation time, depicted as dashed lines in Fig. 3.7, further highlights the efficiency of the CLSVOF approach. It provides a quicker calculation relative to VOF, a finding that contrasts with prior benchmark cases. This performance gap stems from VOF's struggle to meet the convergence criteria at every timestep in this benchmark scenario, necessitating additional iterations for solving the equations, and consequently extending the computation times.

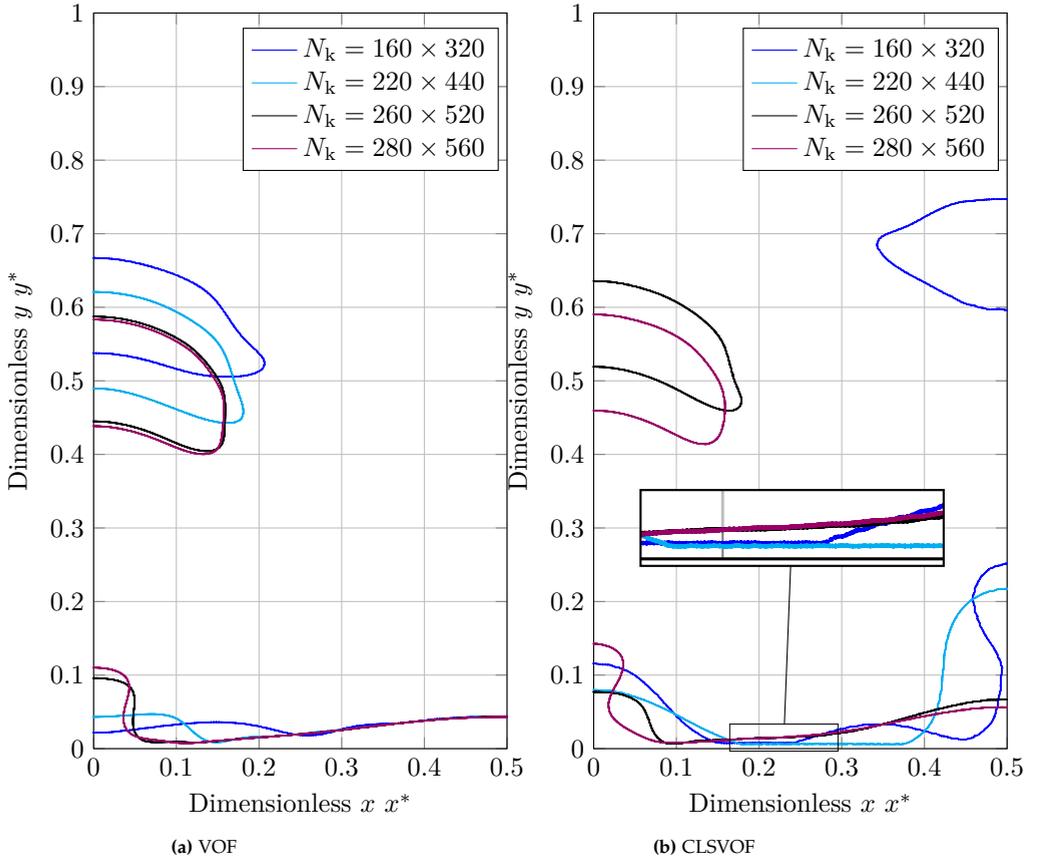


**Fig. 3.7:** Solid lines show the logarithmic representation of the error  $e_\delta$  (equation 2.31) and dashed lines reflect the computation time of VOF and CLSVOF at various grid structures for solving laminar film condensation on a vertical plate benchmark case. The horizontal axis ( $N_k$ ) shows number of grids.

### 3.3.4 2D Film Boiling

To investigate this benchmark case, four grid topologies with  $N_k = [160 \times 320, 220 \times 440, 260 \times 520, 280 \times 560]$  is chosen. Fig. 3.8 presents a comparison of the first detached bubble shape for both VOF and CLSVOF at a specific time. The bubble shape predicted by CLSVOF for the two least refined grid configurations ( $N_k = [160 \times 320, 220 \times 440]$ ) stands out as the first noticeable outcome depicted in Fig. 3.8b. An enlarged view of this graph reveals that the interface is connected to the bottom in these two grid configurations. This trend has not been seen with the VOF as illustrated in Fig. 3.8a, demonstrating that when the mesh configuration is coarse and the interface is near to and parallel to the wall, the CLSVOF approach is unsuccessful. Accordingly, the results of the CLSVOF for the grid structure  $N_k = [160 \times 320, 220 \times 440]$  will not be discussed in the subsequent analysis of the Nusselt number.

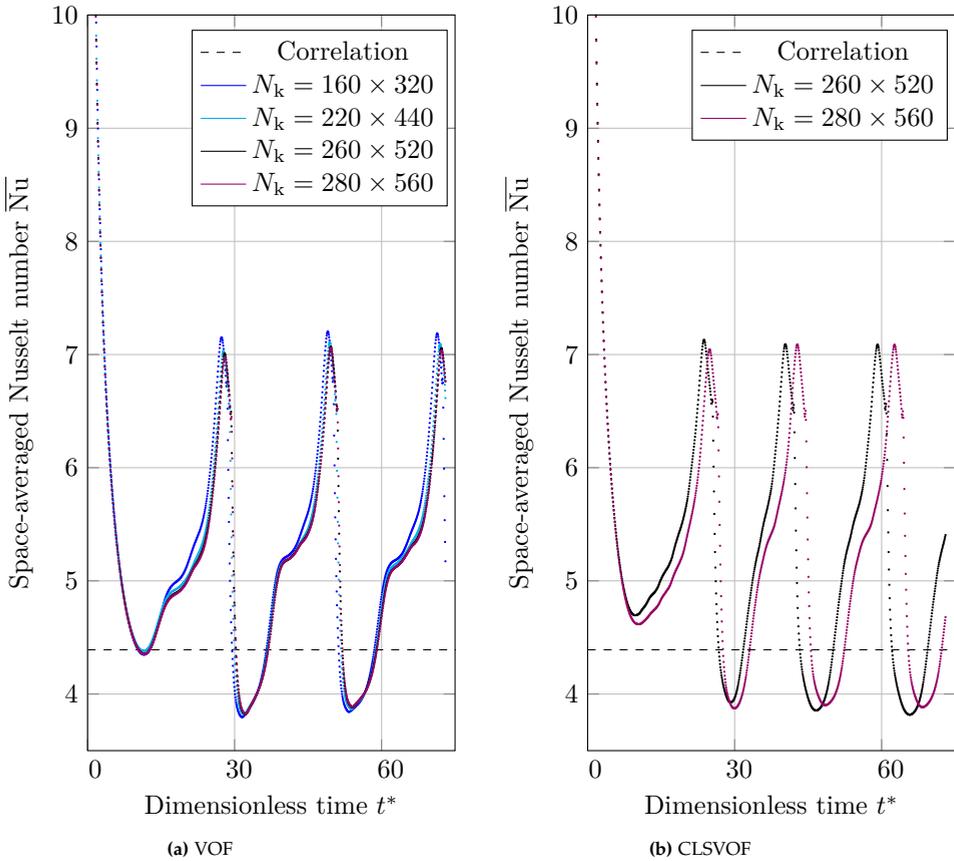
### 3.3. Benchmark Cases



**Fig. 3.8:** A representation of the condensed liquid film interface, obtained by VOF and CLSVOF for 2D film boiling problem at a specified height and different grid sizes ( $N_k = [150 \times 300, 175 \times 350, 200 \times 400, 225 \times 450]$ ). Adopted from Yahyaee et al. [28] © 2022 Licensee MDPI, Basel, Switzerland.

From Fig. 3.8, for the VOF, a grid structure of  $260 \times 520$  can be considered optimal (Grid size  $\approx 4.4 \times 4.4 \times 10^{-6} \text{ mm}^2$ ), while for VOF-MULES, the optimum mesh should be  $280 \times 560$  for modeling the simulation (Grid size  $\approx 4.1 \times 4.1 \times 10^{-6} \text{ mm}^2$ ).

Fig. 3.9a and Fig. 3.9b reveal the changes in the Nusselt number in relation to dimensionless time. The thickness of the film proves to be a significant factor in the Nusselt number. A thin vapour film tends to have a larger heat flux, while the opposite holds true for a thicker film. Consequently, the average Nusselt number and heat flux increase when the vapour swiftly fills the bubble and the remaining layer becomes thinner. However, after the detachment phase, the vapour returns to the superheated wall, causing a decrease in the Nusselt value.



**Fig. 3.9:** The representation of space-averaged Nusselt number, obtained by numerical (VOF and CLSVOF) and Berenson correlation (equation 2.38) for 2D film boiling problem and with different grid sizes. Adopted from Yahyaee et al. [28] © 2022 Licensee MDPI, Basel, Switzerland.

### 3.3.5 Stationary 3D Spherical Bubble Scenario

This section revisits the three-dimensional, gravity-neutral, non-evaporating bubble case, previously explored in the context of VOF-MULES and VOF-isoAdvection methods (Section 2.5.5). The geometry and boundary conditions are the same as those detailed in Fig. 2.18, and are not repeated here for brevity. Fluid properties specific to this case are listed in Table 2.4.

Unwanted non-physical velocities can emerge from a numerical imbalance between the discretization errors in pressure gradient and surface tension forces within the Navier-Stokes equations. Fig. 3.10 shows that both VOF and CLSVOF methods exhibit spurious currents, albeit with varying magnitudes. Specifically, CLSVOF tends to generate a higher amount of these artificial velocities compared to the VOF method.

### 3.3. Benchmark Cases

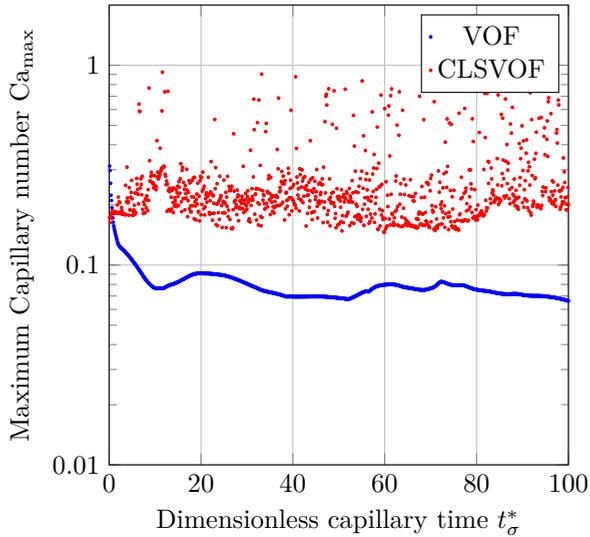


Fig. 3.10: Comparison of maximum capillary numbers for VOF and CLSVOF methods.

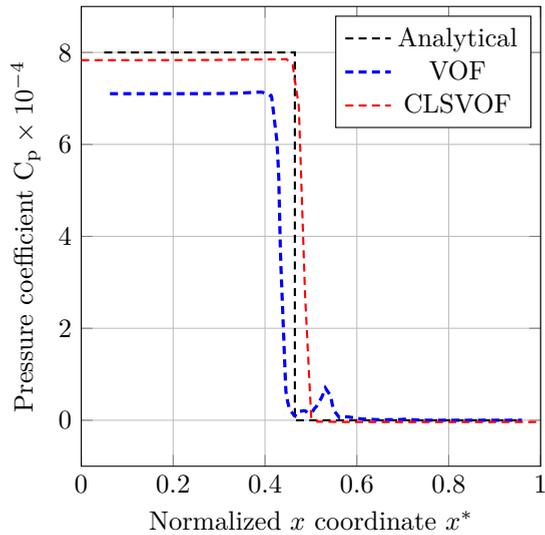
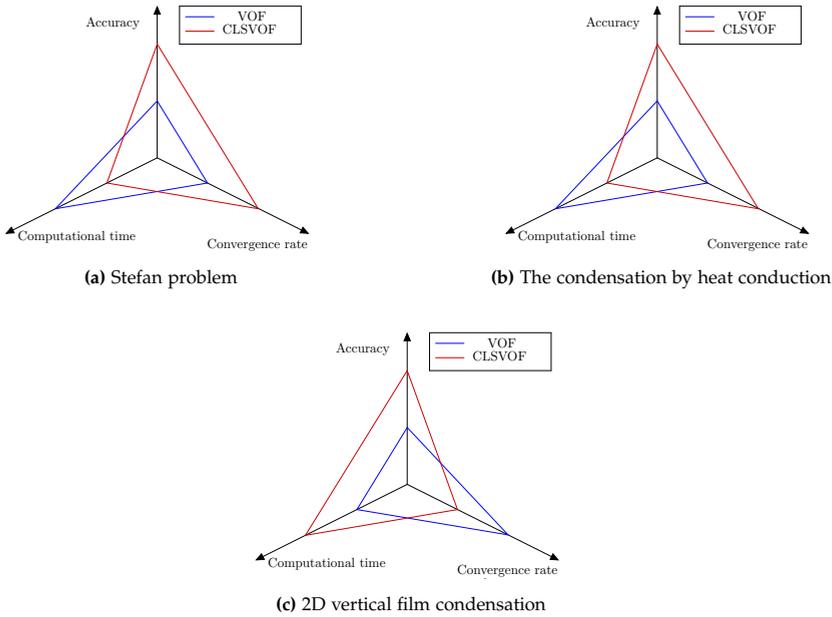


Fig. 3.11: Dimensionless pressure coefficient across the static bubble for VOF and CLSVOF methods.

Another point of interest is the dimensionless pressure coefficient ( $C_p$ ), shown in Fig. 3.11. Unlike the VOF method, CLSVOF shows a better approximation for  $C_p$ , particularly near the interface ( $x^* = 0.5$ ). However, this comes at the cost of increased spurious currents, as indicated earlier.



**Fig. 3.12:** VOF and CLSVOF algorithms' performances in three thermal phase change benchmark instances, shown graphically in terms of accuracy, computing time, and convergence rate. Each axis depicts a study domain and has a pair of values. Methods with better performance in a given region will be assigned a higher value, while those with less performance will be assigned a lower value. Adopted from Yahyae et al. [28] © 2022 Licensee MDPI, Basel, Switzerland.

### 3.4 Conclusion

In the standard thermally driven phase change solvers available in OpenFOAM, the VOF method is commonly used for interface capturing. However, this method may exhibit limitations in accurately computing interface curvature. To address this, a Coupled Level Set-Volume of Fluid (CLSVOF) method is implemented in an OpenFOAM solver<sup>1</sup>. This solver is developed using OpenFOAM version 2006 and integrates the robust features of interPhaseChangeFoam, phaseChangeHeatFoam, and sclsVOFFoam. This modified solver was then rigorously compared to the traditional VOF method using four distinct benchmark cases: the Stefan problem, horizontal film condensation, condensation on a vertical plate, and 2D film boiling.

Additionally, the study is supplemented with a non-thermal benchmark case known as the "Stationary 3D Spherical Bubble Scenario". This case serves as a crucial tool for evaluating curvature prediction and the magnitude of

<sup>1</sup>The solver for the benchmark cases is available at <https://github.com/AAU-OpenFOAM/LSThermalPhaseChangeFlow>

### 3.4. Conclusion

spurious currents. Unlike thermal phase change scenarios, where significant interface motion makes it nearly impossible to isolate and evaluate such currents, the stationary nature of this case allows for a more targeted analysis. The results revealed that while CLSVOF excels in curvature prediction, it also generates higher spurious currents compared to VOF.

Fig. 3.12 offers a summary comparing CLSVOF and VOF in terms of accuracy, computational time, and convergence rate. Generally speaking, CLSVOF provides enhanced accuracy in curvature prediction but at the cost of increased computational time and higher spurious currents. Another observation from 2D film boiling benchmark case is that under specific grid conditions and interface positions, CLSVOF may yield inaccurate representations of film boiling<sup>2</sup>.

---

<sup>2</sup>It should be noted that the study adhered to fixed numerical schemes and solution algorithms for both methods, as outlined in section 2.3. These conditions were kept constant to ensure a fair comparison, and thus the conclusions are applicable within the context of these chosen conditions.

## References

- [1] A. Bourlioux, "A coupled level-set volume-of-fluid algorithm for tracking material interfaces," in *proceedings of the 6th international symposium on computational fluid dynamics, Lake Tahoe, CA*, vol. 15, 1995.
- [2] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations," *Journal of computational physics*, vol. 79, no. 1, pp. 12–49, 1988.
- [3] M. Sussman, "A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow," *Journal of Computational Physics*, vol. 114, no. 1, pp. 146–159, 9 1994.
- [4] F. Losasso, R. Fedkiw, and S. Osher, "Spatially adaptive techniques for level set methods and incompressible flow," *Computers & Fluids*, vol. 35, no. 10, pp. 995–1010, 2006.
- [5] D. L. Chopp, *Computing minimal surfaces via level set curvature flow*. University of California, Berkeley, 1991.
- [6] G. Russo and P. Smereka, "A Remark on Computing Distance Functions," *Journal of Computational Physics*, vol. 163, no. 1, pp. 51–67, 9 2000.
- [7] J. U. Brackbill, D. B. Kothe, and C. Zemach, "A continuum method for modeling surface tension," *Journal of computational physics*, vol. 100, no. 2, pp. 335–354, 1992.
- [8] M. M. Francois, S. J. Cummins, E. D. Dendy, D. B. Kothe, J. M. Sicilian, and M. W. Williams, "A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework," *Journal of Computational Physics*, vol. 213, no. 1, pp. 141–173, 2006.
- [9] Y. Renardy and M. Renardy, "PROST: A Parabolic Reconstruction of Surface Tension for the Volume-of-Fluid Method," *Journal of Computational Physics*, vol. 183, no. 2, pp. 400–421, 12 2002.
- [10] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti, "Modelling Merging and Fragmentation in Multiphase Flows with SURFER," *Journal of Computational Physics*, vol. 113, no. 1, pp. 134–147, 7 1994.
- [11] M. Rudman, "Volume-tracking methods for interfacial flow calculations," *International journal for numerical methods in fluids*, vol. 24, no. 7, pp. 671–691, 1997.
- [12] M. Raessi and H. Pitsch, "Modeling interfacial flows characterized by large density ratios with the level set method," *Center for Turbulence Research Annual Research Briefs*, pp. 159–169, 2009.
- [13] S. Popinet, "An accurate adaptive solver for surface-tension-driven interfacial flows," *Journal of Computational Physics*, vol. 228, no. 16, pp. 5838–5866, 9 2009.
- [14] E. Olsson and G. Kreiss, "A conservative level set method for two phase flow," *Journal of Computational Physics*, vol. 210, no. 1, pp. 225–246, 11 2005.
- [15] M. Sussman and E. G. Puckett, "A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows," *Journal of computational physics*, vol. 162, no. 2, pp. 301–337, 8 2000.

## References

- [16] M. Sussman, "A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles," *Journal of Computational Physics*, vol. 187, no. 1, pp. 110–136, 2003.
- [17] M. Griebel and M. Klitz, "CLSVOF as a fast and mass-conserving extension of the level-set method for the simulation of two-phase flow problems," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 71, no. 1, pp. 1–36, 2017.
- [18] T. Ménard, S. Tanguy, and A. Berlemont, "Coupling level set/VOF/ghost fluid methods: Validation and application to 3D simulation of the primary break-up of a liquid jet," *International Journal of Multiphase Flow*, vol. 33, no. 5, pp. 510–524, 5 2007.
- [19] K. Yokoi, "A practical numerical framework for free surface flows based on CLSVOF method, multi-moment methods and density-scaled CSF model: Numerical simulations of droplet splashing," *Journal of Computational Physics*, vol. 232, no. 1, pp. 252–271, 1 2013.
- [20] M. Arienti and M. Sussman, "An embedded level set method for sharp-interface multiphase simulations of Diesel injectors," *International Journal of Multiphase Flow*, vol. 59, pp. 1–14, 2 2014.
- [21] A. Ferrari, M. Magnini, and J. R. Thome, "A Flexible Coupled Level Set and Volume of Fluid (flexCLV) method to simulate microscale two-phase flow in non-uniform and unstructured meshes," *International Journal of Multiphase Flow*, vol. 91, pp. 276–295, 5 2017.
- [22] M. Dianat, M. Skarysz, and A. Garmory, "A Coupled Level Set and Volume of Fluid method for automotive exterior water management applications," *International Journal of Multiphase Flow*, vol. 91, pp. 19–38, 5 2017.
- [23] S. Omar, "Development of a hyperbolic equation solver and the improvement of the OpenFOAMR two-phase incompressible flow solver," no. December, 2018.
- [24] N. Samkhaniani and M. R. Ansari, "The evaluation of the diffuse interface method for phase change simulations using OpenFOAM," *Heat Transfer - Asian Research*, vol. 46, no. 8, pp. 1173–1203, 2017.
- [25] S. Menon, J. Nagawkar, and H. Nilsson, "Coupled Level-Set with VOF inter-Foam," *Openfoam*, 2016.
- [26] S. Shin and B. Choi, "Numerical simulation of a rising bubble with phase change," *Applied Thermal Engineering*, vol. 100, pp. 256–266, 2016.
- [27] A. H. Rajkotwala, A. Panda, E. A. Peters, M. W. Baltussen, C. W. van der Geld, J. G. Kuerten, and J. A. Kuipers, "A critical comparison of smooth and sharp interface methods for phase transition," *International Journal of Multiphase Flow*, vol. 120, p. 103093, 2019. [Online]. Available: <https://doi.org/10.1016/j.ijmultiphaseflow.2019.103093>
- [28] A. Yahyae, A. S. Bahman, K. Olesen, and H. Sørensen, "Level-Set Interface Description Approach for Thermal Phase Change of Nanofluids," *Nanomaterials*, vol. 12, no. 13, p. 2228, 2022.

## References

## Chapter 4

# Nanofluid Film Boiling

This chapter shifts focus towards enhancing boiling cooling methods, particularly by incorporating nanofluids instead of traditional pure fluids, and refining the numerical methods used for analyzing these processes. The motivation for this shift lies in the remarkable ability of nanofluids to improve heat transfer rates and the scarcity of comprehensive CFD studies in this area.

A notable limitation in most numerical studies on nanofluid boiling is the absence of specific equations to account for nanoparticle concentration, leading to an oversight of crucial aspects of nanoparticle behavior. In cases where these equations are present, they often apply only to either the vapor or the liquid phase, neglecting interactions in the other phase. The absence of a comprehensive equation and model hinders the in-depth analysis of nanoparticle concentration dynamics specifically at the vapor-liquid interface.

Addressing this gap, the research utilizes a foundational governing equation for nanoparticle concentration, as referenced in several notable studies [1–5]. This established equation, traditionally applied to a single phase (either vapor or liquid), forms the basis of our computational model. However, the model's scope is extended to encompass both the vapor and liquid phases, as well as the interface between them. To achieve this, the Continuous-Species-Transfer (CST) method, introduced by Marschall et al. [6], is employed.

Marschall et al. have demonstrated the application of the CST method to extend a single-phase concentration equation across the entire domain, including both phases and their interface. A key component of this method is the use of the Henry constant, which correlates the concentrations in different phases by considering the solubilities of various species. The current research adopts the CST method, applying it to derive a governing equation for nanoparticle concentration across the entire domain, including vapor, liquid, and the interface. This approach effectively addresses the first identified

gap in nanofluid film boiling simulations.

The effectiveness of the discussed approach is showcased by simulating two-dimensional axisymmetric film boiling of water- $\text{Al}_2\text{O}_3$  nanofluid on a vertical cylinder. The results from this simulation align with existing analytical models. A notable achievement of this model is its ability to be the first CFD model to simulate nanoparticle concentration at the vapor-liquid interface during the evaporation of the vapor film. The analysis conducted in this study includes examining nanoparticle concentration and temperature distributions, along with assessing changes in the thermophysical properties of the nanofluids during film boiling. Additionally, the research explores the relationship between nanoparticle concentration and heat transfer efficiency in nanofluid boiling, specifically looking at how variations in nanoparticle concentration might influence the space-averaged Nusselt number, a key indicator of heat transfer efficiency.

In Chapter 4, the text is organized as follows: Section 4.1 provides a comprehensive Literature Review on the Boiling of Nanofluids, identifying gaps in existing research and outlining the novel contributions of this study. Section 4.2 delves into the Mathematical Framework and Governing Equations that underpin the research. Following this, Section 4.3 describes the Problem (two-dimensional axisymmetric film boiling of water- $\text{Al}_2\text{O}_3$  nanofluid on a vertical cylinder), including detailed information on the geometry, boundary conditions, and initial conditions employed in the study. Section 4.4 presents the Results and Discussion, where the findings of the simulations and analyses are explored in depth. Finally, Section 4.5 concludes the chapter, summarizing key insights and implications of the research conducted.

## 4.1 Literature Review On Boiling Of Nanofluids

### Relevance Of Nanofluid Boiling Research

The exploration of heat transfer enhancement through nanofluid boiling has garnered significant attention in various sectors, including automotive engineering, power generation, advanced electronic cooling, aircraft environmental control, and even in the thermal management of spacecraft [7–11]. A key area of focus within this field is film boiling in nanofluids, an emerging and relatively unexplored research domain. This type of boiling is especially crucial in applications subjected to extreme thermal conditions where effective thermal management and safety are of utmost importance [12–14]. The introduction of nanoparticles into boiling fluids introduces new challenges and complexities, thus drawing significant interest in both theoretical and experimental fluid dynamics studies [15–17].

Building on Maxwell's foundational work [18], it is understood that dis-

persing fine particles in base liquids like water, oils, or alcohols can significantly enhance heat transfer. However, the use of microparticles has been hindered by issues such as erosion, clogging, and sedimentation [19]. In response, Choi et al. [20] introduced the concept of nanofluids, which are fluids containing nanoparticles uniformly distributed within them. Nanofluids display unique thermal properties that could offer substantial advantages over traditional fluids, such as improved thermal conductivity and changes in viscosity and surface tension [21, 22]. However, integrating nanoparticles reduces the effective heat capacity. Consequently, careful consideration is required when incorporating nanofluids into thermal management systems, keeping in mind the specific requirements and goals of the application.

##### **Influence of Nanoparticles on Surface Dynamics in Nanofluid Boiling**

The role of nanoparticles in modifying surface interactions and fluid dynamics during nanofluid boiling is significant. A key aspect of this influence is observed on heated surfaces, where the deposition of nanoparticles alters crucial properties like surface wettability, heat transfer coefficients, and capillary forces [23–25]. This deposition results in a layer that enhances the effective contact area between the liquid and solid phases, thereby facilitating more efficient heat transfer during phase changes [26]. Furthermore, the presence of nanoparticles plays a pivotal role in determining surface wettability, which in turn affects the boiling process. Factors such as the radius of the triple line and the contact angle are governed by the interaction of forces at the solid-liquid-gas interface [27, 28]. Nanoparticle composition and concentration are instrumental in modulating surface tension and contact angles. For instance, silica-based nanofluids demonstrate lower surface tension compared to water, impacting boiling characteristics including bubble departure size [29–31].

Nanoparticles significantly change how droplets evaporate and bubbles form, and this change varies with the size and type of the nanoparticles. Smaller nanoparticles tend to spread out more evenly, while larger ones gather mainly at the edges, affecting how evaporation happens [32]. Gold nanoparticles, for example, have been seen to hold the triple line (where liquid, gas, and solid meet) in place, strongly influencing how bubbles behave [33]. In contrast, with some nanofluids, the triple line can stretch a lot, up to 50 times the size of the nanoparticles, because of increased pressure effects at this junction. These effects change both how the liquid spreads and how bubbles are formed [34–36].

##### **CFD Studies On Nanofluid Boiling Without Considering Governing Equation For Nanoparticles Concentration**

In the last decade, the field of CFD has seen a notable increase in research focused on enhancing heat transfer efficiencies in nanofluid boiling applica-

tions. Abedini et al. [37] conducted an analysis on the flow boiling characteristics of subcooled nanofluids, revealing that a nanoparticle concentration between 1% and 2% yielded the most efficient heat transfer, surpassing the performance of higher concentrations. As research progressed, Shoghl et al. [38] combined experimental techniques with CFD modeling to scrutinize bubble behavior in nanofluid pool boiling. Their study underscored the significance of porous layers and surfactants in the process.

Similarly, Qi et al. [19] explored the influence of TiO<sub>2</sub> nanoparticles within water-based nanofluids, verifying enhancements in both flow resistance and heat transfer properties. The work of Abdollahi et al. [39] delved into the boiling flows in micro-channels under laminar conditions, shedding light on the effects of nanofluid type, nanoparticle density, and particle dimensions on both heat transfer and fluid dynamics.

Alongside these investigations, Mohammed et al. [40, 41] made substantial contributions in 2018 and 2020, applying CFD to a variety of boiling scenarios, ranging from salt solution concentrations to intricate nanofluid flows in rectangular channels. Their findings echoed previous research, underscoring that an increase in nanoparticle concentration significantly enhances heat transfer rates and affects vapor pressure dynamics.

In their research, Soleimani et al. [42] concentrated on the subcooled boiling characteristics of HFE-7100 nanofluids. Their findings indicated a modest improvement in heat transfer compared to two-phase flow boiling, suggesting nanofluids' viability for thermal management applications. Li et al. [43] broadened the investigation by examining the boiling behavior under different gravitational conditions, shedding light on the variability of Heat Transfer Coefficients (HTCs) due to external factors.

More recent studies by Zaboli et al. [44] and Alomar et al. [45] have focused on the impact of silica and salt solution nanoparticles in nanofluids. Zaboli et al. discovered that even a small nanoparticle volume fraction of 0.1% could significantly improve heat transfer, particularly in high thermal flux scenarios. Alomar et al., utilizing a commercial CFD framework, analyzed various phases of concentrated salt solution nanofluids and deduced that higher nanoparticle concentrations markedly improve heat transfer coefficients. Concurrently, Mavi et al. developed advanced numerical models to simulate three-phase boiling and bubble formation in rectangular channels, validating that nanoparticles play a crucial role in enhancing both heat transfer and boiling behavior [46].

## **CFD Studies On Nanofluid Boiling Considering Governing Equation For Nanoparticles Concentration**

Buongiorno's pivotal work [1] discussed the primary mechanisms driving nanoparticle transport in nanofluids, pinpointing Brownian diffusion and

#### 4.1. Literature Review on Boiling of Nanofluids

thermophoresis as key influencers. Brownian motion involves the erratic movement of particles due to collisions with fast-moving molecules in the fluid, while thermophoresis describes particle migration in response to temperature gradients. Notably, the prior CFD studies mentioned do not integrate these transport phenomena into their models, nor do they include a specific equation to govern nanoparticle motion.

Nanoparticle mobility plays a critical role in enhancing both the heat transfer efficiency and the thermal attributes of nanofluids. Echoing Buongiorno's insights [1], Brownian diffusion and Thermophoresis are key mechanisms influencing nanoparticle dynamics in these fluids. The contribution of Brownian diffusion to nanoparticle transportation in nanofluids has been a focal point of many studies. Yang et al. [2] revisited the Buongiorno model, examining how nanoparticle migration affects heat transfer in alumina-water and titania-water nanofluids within cylindrical annular spaces. Furthering this line of inquiry, Malvandi and colleagues [47] explored the combined effects of buoyancy and nanoparticle migration in vertical annular ducts. In a similar study, Malvandi and Ganji [3] investigated alumina-water nanofluids in flat channels, noting the varied nanoparticle concentrations along the temperature gradient, leading to non-uniform concentration profiles. They highlighted the unique heat transfer characteristics when Brownian motion is predominant. Hedayati and Domairry's research [48, 49] demonstrated the significant influence of nanoparticle migration on heat transfer in titania-water nanofluids, both in horizontal and vertical channels.

Michaelides [50] expanded the research scope, analyzing the averaged behaviors of nanoparticles in various base fluids including water, ethyl glycol, engine oil, and R134a, confirming the significance of thermophoresis and Brownian movement. Malvandi et al. [17] conducted theoretical research on the impact of thermophoresis and Brownian motion in enhancing heat transfer during film boiling of nanofluids over a vertical cylinder, observing that smaller nanoparticles tend to accumulate near the heated surface, thereby boosting the heat transfer rate. They also found that alumina nanoparticles offered better cooling efficiency than titania nanoparticles. Lin and Jiang [5] studied nanofluids in a rotating circular groove, incorporating Brownian motion and thermophoresis in their finite element analysis. Their results underscored the importance of thermophoresis and nanoparticle thermal conductivity on heat transfer enhancement, noting a greater efficiency in the Maxwell Model compared to the Traditional Model.

It is crucial to note, however, that these studies have predominantly concentrated on the vapor film aspects, often overlooking the nanofluid liquid phase in contact with the vapor film. This oversight creates a gap in current research, as the governing equations are usually solved for the vapor film alone, without adequately addressing the nanoparticle transfer between the vapor and liquid phases of the nanofluid.

## Identifying Gaps And Novel Contributions

As identified in the study by Buongiorno [1], Brownian diffusion and thermophoresis are among the factors that play significant roles in governing nanoparticle movement in nanofluids. However, the various numerical studies referenced [19, 37–46] do not account for these mechanisms, nor do they include a specific equation for nanoparticle motion. This approach often leads to an underestimation of the nanofluid's heat transfer capabilities [51].

On the other hand, a smaller subset of literature [2, 3, 5, 17, 47–50] does incorporate equations specifically addressing nanoparticle motion. However, these studies primarily focus on the vapor film domain, while not fully addressing the interactions within the liquid nanofluid phase adjacent to this vapor layer. A thorough understanding of how this vapor film interacts with the adjacent liquid nanofluid phase and how nanoparticle concentration varies through the vapor-liquid interface is vital for comprehensively evaluating the system's hydraulic and thermal behavior. Consequently, there is a notable gap in current research: existing models and analyses primarily focus on the vapor film, resulting in an inadequate exploration of how nanoparticle concentration varies between the vapor and liquid phases during nanofluid boiling.

To address the mentioned research gap, this study makes use of a well-established governing equation for nanoparticle concentration, which is cited in a range of literature [1–5]. Typically applied to single-phase scenarios (either in vapor or liquid form), this equation forms the core of the computational framework developed here. The research ambitiously expands this model to include both vapor and liquid phases and their interfacial interactions, utilizing the Continuous-Species-Transfer (CST) approach pioneered by Marschall et al. [6].

In their work, Marschall et al. have effectively illustrated how the CST method can be adapted to expand a single-phase concentration equation to cover the entire domain, spanning both phases and the interface. The integration of the Henry constant is pivotal in this methodology, as it facilitates the correlation of species concentrations between phases based on solubility differences. This study leverages the CST approach to formulate a comprehensive governing equation for nanoparticle concentration that encompasses the vapor phase, liquid phase, and the interface, addressing a significant gap in the field of nanofluid film boiling simulations.

## 4.2 Mathematical Framework And Governing Equations

This study presents a model designed to solve complex issues in identifying the vapor-liquid interface in boiling nanofluids. This model combines Volume-of-Fluid (VOF)<sup>1</sup> and pseudo-VOF techniques to control nanoparticle movement between nanofluid and vapor phases. By using Henry's Law, changes in concentration are addressed across vapor-liquid interface cells. The Continuous-Surface-Force (CSF) Method, referenced in [52], has proven effective for dealing with boundary issues. Building on CSF, the Continuous-Species-Transfer (CST) Method, described in [6], accurately models species movement across changing fluid interfaces. This paper adds to these methods by using models of Brownian motion and thermophoresis to show how nanoparticles move in nanofluids.

### 4.2.1 Theoretical Framework And Methodology

The core idea of this study's solver for tracking nanoparticle concentration changes across liquid nanofluid and its vapor interface is based on the concept of 'immersed interface', as discussed in [53]. This means treating the boundary where liquid nanofluid meets its vapor as part of the entire area being analyzed. Consequently, the same mathematical rules apply across the liquid nanofluid, evaporated part, and their boundary.

The governing equations are derived starting from fundamental conservation laws applicable to a single fluid phase. 'Conditional volume-averaging' is then employed, as outlined in [54–56]. This technique involves initially tailoring equations to a specific fluid phase using a phase-indicator function, labeled as "lnf" for liquid nanofluid and "vnf" for the evaporated volume [6]:

---

<sup>1</sup>In Chapters 2 and 3, the study focused on implementing and evaluating the VOF-isoAdvector and CLSVOF methods, highlighting several advantages these methods have over the traditional VOF approach. However, in this chapter, the VOF method is still utilized. The reason behind this choice stems from the fact that isoAdvector and CLSVOF in chapters 2 and 3 were developed on a solver foundation laid by the interPhaseChangeFoam solver. Conversely, the current chapter requires the implementation of specific libraries to model thermophysical properties and their dependencies on temperature and nanoparticle concentration. These essential libraries are available in the icoReactingMultiPhaseInterFoam solver. Unfortunately, the integration of these thermophysical property calculations for nanofluids into the previously developed thermal phase change solvers for VOF-isoAdvector and CLSVOF was not feasible.

Given that the primary objective of this chapter is to address the existing gaps in nanofluid boiling research, and considering that VOF, VOF-isoAdvector, and CLSVOF belong to different categories of discussion, the new solver and associated libraries have been implemented based on the OpenFOAM v2006's icoReactingMultiPhaseInterFoam solver, which relies on the VOF method.

$$I_{\text{Inf}}(x, t) = \begin{cases} 1 & \text{if Inf phase is present at } (x, t) \\ 0 & \text{otherwise} \end{cases}. \quad (4.1)$$

The process of 'volume-averaging' simplifies as follows for any flow property, denoted by  $\Phi$  [6]:

$$\overline{I_{\text{Inf}}\Phi} = \frac{1}{V} \int_V I_{\text{Inf}}(\mathbf{x} + \boldsymbol{\eta}, t) \Phi(\mathbf{x} + \boldsymbol{\eta}, t) d\mathbf{x}_{\boldsymbol{\eta}}. \quad (4.2)$$

Here, the calculation spans a fixed control volume  $V$ , with  $\mathbf{x}$  as its center and  $\boldsymbol{\eta}$  representing any point within  $V$ .

Combining these averaged equations for both nanofluid and vapor phases forms a comprehensive set of governing equations. They align with the 'immersed interface' concept, applicable to the entire area of study, including the nanofluid, vapor, and their interface.

This approach clarifies the role of the volume-averaged indicator function, which indicates the fraction of the control volume occupied by each phase. For instance, with  $\Phi = 1$ :

$$\overline{I_{\text{Inf}}(\mathbf{x}, t)} \equiv \alpha_{\text{Inf}} = \begin{cases} 1 & \text{completely in liquid nanofluid phase,} \\ (0, 1) & \text{at the phase boundary,} \\ 0 & \text{completely in vapor nanofluid phase.} \end{cases} \quad (4.3)$$

## 4.2.2 Bubble Dynamics Via The Volume-Of-Fluid Method

### Fundamental Equations

The main equations that describe the behavior of the two-phase system, which includes changes in the thermal phase, are based on the principles of volume-averaged conservation laws. These laws are crucial for understanding the flow in this two-phase system that experiences thermal phase transitions.

The mass conservation equation is:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0, \quad (4.4)$$

This equation is essential for making sure mass is conserved in the system.

The equation for keeping track of momentum is:

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \mathbf{f}_g + \mathbf{f}_\sigma, \quad (4.5)$$

The equation that shows how the phase fraction  $\alpha_{\text{Inf}}$  changes is [57]:

## 4.2. Mathematical Framework and Governing Equations

$$\frac{\partial \alpha_{\text{lnf}}}{\partial t} + \nabla \cdot (\mathbf{U} \alpha_{\text{lnf}}) + \nabla \cdot (\mathbf{U}_r \alpha_{\text{lnf}} (1 - \alpha_{\text{lnf}})) = -\dot{m}''' \left[ \frac{1}{\rho_{\text{lnf}}} - \alpha_{\text{lnf}} \left( \frac{1}{\rho_{\text{lnf}}} - \frac{1}{\rho_{\text{vnf}}} \right) \right], \quad (4.6)$$

where  $\dot{m}'''$  is the volumetric mass transfer rate which will be calculated further. The viscous stress tensor,  $\boldsymbol{\tau}$ , for incompressible fluids, is given by  $\boldsymbol{\tau} = -\mu (\nabla \mathbf{U} + \nabla \mathbf{U}^T)$ . The forces due to gravity and surface tension at interfaces are shown by  $\mathbf{f}_g$  and  $\mathbf{f}_\sigma$ , respectively. In addition,  $\alpha_{\text{lnf}}$  is the phase fraction for the liquid nanofluid, with the vapor phase being its complement to make up the two-phase system.

In both the momentum equation (equation 4.5) and the energy equation (equation 4.38), the variables such as density  $\rho$ , viscosity  $\mu$ , thermal conductivity  $\lambda$ , and specific heat at constant pressure  $c_p$  are treated as combined properties in the interface area. The density  $\rho$  of the mixture is calculated using the volumetric phase fractions as follows:

$$\rho = \sum_{k=\text{lnf},\text{vnf}} \alpha_k \bar{\rho}^k. \quad (4.7)$$

The other mixture properties are determined in a manner similar to the calculation of mixture density.

### Closure Models for Governing Equations

The use of conditional volume-averaging in equation development results in certain terms being undefined. Consequently, the governing equations are incomplete and necessitate additional models for a full description.

**Surface Tension Interface Model:** Surface tension's impact on momentum transfer is significant, particularly in the finite-width transition zone at the interface. Although surface tension acts as a surface force, it introduces a momentum discontinuity at this interface.

For addressing this, the Continuous-Surface-Force (CSF) method by [52] is utilized. The CSF approach is defined by the equation:

$$\mathbf{f}_\sigma = \sigma \kappa \nabla \alpha_{\text{lnf}} \quad \text{where} \quad \kappa = -\nabla \cdot \left( \frac{\nabla \alpha_{\text{lnf}}}{|\nabla \alpha_{\text{lnf}}|} \right), \quad (4.8)$$

Here,  $\sigma$  represents the constant surface tension coefficient. This model does not account for variable surface tension effects such as Marangoni effects.

**Model for Interphase Relative Velocity:** In pseudoVOF methods, accurately locating the interface between different fluid phases is crucial for the simulation's precision. This accuracy helps in reducing errors due to numerical diffusion. Notably, the third component in the equation for phase fraction (equation 4.6) only has a value in the interface areas, due to the existence of the term  $\alpha_{\text{lnf}}(1 - \alpha_{\text{lnf}})$ .

Drawing from the studies by [58] and [59], a key method to keep the interface sharp involves treating the relative velocity  $\mathbf{U}_r$  as a "compressive velocity"  $\mathbf{U}_c$ , directed perpendicular to the phase boundary. Its mathematical form is:

$$\mathbf{U}_c = \min [c_\alpha |\mathbf{U}|, \max(|\mathbf{U}|)] \frac{\nabla \alpha_{\text{Inf}}}{|\nabla \alpha_{\text{Inf}}|}, \quad (4.9)$$

where  $c_\alpha$  typically lies between 1 and 4. By applying this approach, the third component in the phase fraction equation (equation 4.6) turns into a "compressive term." This part helps in maintaining a clear boundary between the phases, positioning itself perpendicular to the interface, without favoring any specific fluid phase.

### 4.2.3 Modeling Nanoparticle Dynamics With CST, Brownian Motion, And Thermophoresis

This research addresses the behavior of nanoparticles in fluids using a primary equation that describes the volumetric concentration of these particles in both vapor and nanofluid states [5, 17, 50]. This equation is:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \mathbf{U}) = \nabla \cdot (D_B \nabla \phi) + \nabla \cdot \left[ D_{\text{Th}} \frac{\nabla T}{T_\infty} \right]. \quad (4.10)$$

Here,  $\phi$  denotes the volumetric concentration of nanoparticles in a single phase,  $T_\infty$  represents the critical temperature, and  $D_B$  and  $D_{\text{Th}}$  are diffusion coefficients for Brownian motion and thermophoresis, respectively. These coefficients are defined as:

$$D_B = \frac{k_{\text{Bo}} T}{3\pi \mu_{\text{bf}} d_p}, \quad (4.11)$$

$$D_{\text{Th}} = \beta \frac{\mu_{\text{bf}}}{\rho_{\text{bf}}}, \quad (4.12)$$

Key variables in these equations, like  $k_{\text{Bo}}$ ,  $\mu_{\text{bf}}$ ,  $T$ ,  $d_p$ , and  $\rho_{\text{bf}}$ , represent essential physical constants such as Boltzmann constant ( $= 1.3806488 \times 10^{-23} \text{ m}^2 \text{ kg} / \text{ s}^2 \text{ K}$ ), dynamic viscosity, temperature, nanoparticle diameter, and fluid density. The term  $\beta$  is defined as  $0.26 \lambda_{\text{bf}} / (2 \lambda_{\text{bf}} + \lambda_{\text{np}})$ .

These coefficients can be reformulated as:

$$D_B = D'_B T, \quad (4.13)$$

$$D_{\text{Th}} = D'_{\text{Th}} T_\infty \phi, \quad (4.14)$$

The integration of  $D'_B$  and  $D'_{\text{Th}}$  into the governing equation results in the following:

## 4.2. Mathematical Framework and Governing Equations

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \mathbf{U}) = \nabla \cdot (D'_B T \nabla \phi) + \nabla \cdot [D'_{Th} \phi \nabla T]. \quad (4.15)$$

Specific conditions, often referred to as "jump conditions", are introduced:

$$\|\phi\| = \phi_{I,vnf} \cdot (1 - He) \Leftrightarrow He = \frac{\phi_{I,lnf}}{\phi_{I,vnf}}, \quad (4.16)$$

where  $\|\cdot\|$  indicates a discontinuity at the interface, defined as  $\|f\| = f_{I,vnf} - f_{I,lnf}$ . Furthermore, the disparity in nanoparticle solubility between the vapor and liquid nanofluid phases necessitates consideration of the concentration difference across these phases. This disparity is traditionally addressed through Henry's law, with  $He$  as the distribution coefficient. This coefficient often varies from unity, reflecting the differential solubility of nanoparticles in the liquid nanofluid and vapor phases [6]. The application of the "conditioning and volume-averaging" technique further streamlines equation 4.15:

$$\overline{I_{lnf} \frac{\partial \phi}{\partial t}} + \overline{I_{lnf} \nabla \cdot (\phi \mathbf{U})} = \overline{I_{lnf} \nabla \cdot (D'_B T \nabla \phi)} + \overline{I_{lnf} \nabla \cdot (D'_{Th} \phi \nabla T)}. \quad (4.17)$$

This method results in more concise expressions for the concentration of nanoparticles within both the nanofluid and vapor phases:

$$\begin{aligned} \frac{\partial \alpha_{lnf} \bar{\phi}^{lnf}}{\partial t} + \nabla \cdot (\alpha_{lnf} \bar{\phi}^{lnf} \bar{\mathbf{U}}^{lnf}) &= \nabla \cdot (\alpha_{lnf} \overline{D'_B}^{lnf} \overline{T}^{lnf} \nabla \bar{\phi}^{lnf}) \\ &+ \nabla \cdot (\alpha_{lnf} \overline{D'_{Th}}^{lnf} \bar{\phi}^{lnf} \nabla T^{lnf}), \end{aligned} \quad (4.18)$$

$$\begin{aligned} \frac{\partial \alpha_{vnf} \bar{\phi}^{vnf}}{\partial t} + \nabla \cdot (\alpha_{vnf} \bar{\phi}^{vnf} \bar{\mathbf{U}}^{vnf}) &= \nabla \cdot (\alpha_{vnf} \overline{D'_B}^{vnf} \overline{T}^{vnf} \nabla \bar{\phi}^{vnf}) \\ &+ \nabla \cdot (\alpha_{vnf} \overline{D'_{Th}}^{vnf} \bar{\phi}^{vnf} \nabla T^{vnf}) \end{aligned} \quad (4.19)$$

In this research, a unified approach is developed by combining equation 4.18 and equation 4.19 to effectively integrate the concept of an immersed interface. Notably, the original formulations of these equations included surface integral terms. However, in the particular case of film boiling on a vertical cylinder, as addressed in this study, the vapor's motion is primarily driven by gravitational forces, as referenced in [15, 17], and the velocity induced by thermal phase change is negligible compared to velocity caused by gravitational forces. Consequently, these surface integral terms are considered to be of minimal significance and are therefore excluded from the analysis. The study progresses to introduce a parameter for the nanofluid concentration

that spans the entire computational domain and not just in either the liquid nanofluid or vapor phase:

$$\varphi = \alpha_{\text{lnf}} \bar{\phi}^{\text{lnf}} + (1 - \alpha_{\text{lnf}}) \bar{\phi}^{\text{vnf}}. \quad (4.20)$$

This formulation is specifically applicable to terms on the left-hand side of equation 4.18. By writing the concentration governing equations using  $\varphi$  instead of  $\phi$ , one single equation can be used for all domains of liquid nanofluid, vapor, and their interface.

**Brownian Dynamics Term:** Regarding the first term on the right-hand side of equation 4.18, this term should be expressed exclusively in terms of the volumetric nanofluid concentration  $\varphi$  across the entire domain. This term is thus divided into three distinct components:

$$\begin{aligned} \nabla \cdot \left( \sum_{k=\text{lnf},\text{vnf}} \nabla \cdot (\alpha_k \bar{D}'_B{}^k \bar{T}^k \nabla \bar{\phi}^k) \right) &= \nabla \cdot \underbrace{\left( \sum_{k=\text{lnf},\text{vnf}} \nabla \left( \bar{D}'_B{}^k \bar{T}^k \alpha_k \bar{\phi}^k \right) \right)}_{(a)} \\ &- \underbrace{\nabla \cdot \left( \sum_{k=\text{lnf},\text{vnf}} \bar{D}'_B{}^k \bar{T}^k \bar{\phi}^k \nabla \alpha_k \right)}_{(b)} \\ &- \underbrace{\nabla \cdot \left( \sum_{k=\text{lnf},\text{vnf}} \bar{D}'_B{}^k \alpha_k \bar{\phi}^k \nabla \bar{T}^k \right)}_{(c)}. \end{aligned} \quad (4.21)$$

(a) In addressing term *a* within equation 4.21, this study adopts an approximation similar to that proposed by [6]. They defined the molar mixture concentration  $S_i$  as:

$$S_i = \frac{\bar{D}_i^\omega \alpha_\omega \bar{s}_i^\omega + \bar{D}_i^\epsilon \alpha_\epsilon \bar{s}_i^\epsilon}{\alpha_\omega \bar{D}_i^\omega + \alpha_\epsilon \bar{D}_i^\epsilon}, \quad (4.22)$$

where  $\omega$  and  $\epsilon$  represent different phases, and  $\bar{s}_i^\omega$  and  $\bar{s}_i^\epsilon$  are the concentrations of species *i* in these respective phases. The term in the denominator is succinctly represented as the molecular diffusivity  $D_i$ , similar in concept to momentum diffusivity  $\mu$ . Utilizing this approximation, the volumetric concentration  $\varphi$  in the current study is defined as:

$$\varphi \approx \frac{\bar{D}_B^{\text{lnf}} \alpha_{\text{lnf}} \bar{\phi}^{\text{lnf}} + \bar{D}_B^{\text{vnf}} \alpha_{\text{lnf}} \bar{\phi}^{\text{vnf}}}{\alpha_{\text{lnf}} \bar{D}_B^{\text{lnf}} + \alpha_{\text{vnf}} \bar{D}_B^{\text{vnf}}}. \quad (4.23)$$

## 4.2. Mathematical Framework and Governing Equations

where the denominator is concisely rewritten as  $D_B \equiv \alpha_{\text{lnf}} \overline{D_B}^{\text{lnf}} + \alpha_{\text{vnf}} \overline{D_B}^{\text{vnf}}$ . In light of equation 4.13, this approximation (equation 4.23) can be reformulated as:

$$\varphi = \frac{\overline{D_B}^{\text{lnf}} \overline{T}^{\text{lnf}} \alpha_{\text{lnf}} \overline{\phi}^{\text{lnf}} + \overline{D_B}^{\text{vnf}} \overline{T}^{\text{vnf}} \alpha_{\text{lnf}} \overline{\phi}^{\text{vnf}}}{D_B}. \quad (4.24)$$

In cells completely filled with either nanofluid or vapor, the temperature values  $\overline{T}^{\text{lnf}}$  and  $\overline{T}^{\text{vnf}}$  correspond to the average temperature  $T$ , following the mixture equation  $T_{\text{mixture}} = \alpha_{\text{phase1}} \overline{T}^{\text{phase1}} + \alpha_{\text{phase2}} \overline{T}^{\text{phase2}}$  as described in [60]. It is important to note, however, that  $\overline{T}^{\text{lnf}}$  and  $\overline{T}^{\text{vnf}}$  do not necessarily equate to  $T$  at the interface in all cases. In the particular scenario of film boiling on a vertical plate being analyzed here, where both nanofluid liquid and vapor initially reach the saturation temperature and phase change is triggered by a slightly elevated wall temperature ( $\nabla T$  of about 5 degrees), assuming  $\overline{T}^{\text{lnf}} = \overline{T}^{\text{vnf}}$  is a valid approach. Consequently,  $T$  can be considered representative of the temperatures of both the nanofluid liquid and vapor. Assuming  $D'_B$ , defined as  $D'_B = \alpha_{\text{lnf}} \overline{D_B}^{\text{lnf}} + \alpha_{\text{vnf}} \overline{D_B}^{\text{vnf}}$ , similar to the momentum diffusivity  $\mu$ , the denominator term in equation 4.24 simplifies to:

$$\alpha_{\text{lnf}} \overline{D_B}^{\text{lnf}} \overline{T}^{\text{lnf}} + \alpha_{\text{vnf}} \overline{D_B}^{\text{vnf}} \overline{T}^{\text{vnf}} = D'_B T, \quad (4.25)$$

allowing equation 4.24 to be reformulated as:

$$\overline{D_B}^{\text{lnf}} \overline{T}^{\text{lnf}} \alpha_{\text{lnf}} \overline{\phi}^{\text{lnf}} + \overline{D_B}^{\text{vnf}} \overline{T}^{\text{vnf}} \alpha_{\text{vnf}} \overline{\phi}^{\text{vnf}} = D'_B T \varphi. \quad (4.26)$$

This adjustment permits the rewriting of term (a) in equation 4.21 using  $\varphi$ :

$$\begin{aligned} \sum_{k=\text{nf,v}} \nabla \cdot \left( \nabla \left( \overline{D_B}^k \overline{T}^k \alpha_k \overline{\phi}^k \right) \right) \\ = \nabla \cdot \left( \nabla (D'_B T \varphi) \right) \\ + \nabla \cdot (D'_B T \nabla \varphi) + \nabla \cdot (\varphi T \nabla D'_B) + \nabla \cdot (\varphi D'_B \nabla T). \end{aligned} \quad (4.27)$$

(b) The subsequent term in equation 4.21 pertains to the dynamics at the interface. With the relationships  $\alpha_{\text{lnf}} = 1 - \alpha_{\text{vnf}}$  and  $\nabla \alpha_{\text{lnf}} = -\nabla \alpha_{\text{vnf}}$ , this term can be formulated as:

$$\nabla \cdot \left[ \left( \overline{D_B}^{\text{lnf}} \overline{T}^{\text{lnf}} \overline{\phi}^{\text{lnf}} - \overline{D_B}^{\text{vnf}} \overline{T}^{\text{vnf}} \overline{\phi}^{\text{vnf}} \right) \nabla \alpha_{\text{lnf}} \right]. \quad (4.28)$$

In the context of equation 4.28, and referring to the explanation for term (a), it is justifiable to use  $\overline{T}^{\text{lnf}} = \overline{T}^{\text{vnf}}$  as an approximation. Under this assumption,  $T$  effectively represents the temperature for both the nanofluid

liquid and vapor phases. By invoking Henry's law and applying the average jump condition at the interface, equation 4.28 can be modified to:

$$\nabla \cdot \left[ \left( \frac{\overline{D'_B}^{\text{lnf}}}{\overline{D'_B}^{\text{lnf}}} - \frac{\overline{D'_B}^{\text{vnf}}}{He} \right) T \overline{\phi}^{\text{lnf}} \nabla \alpha_{\text{lnf}} \right]. \quad (4.29)$$

Further application of Henry's law, in conjunction with equation 4.20, leads to an expression for a unified concentration  $\varphi$ :

$$\nabla \cdot \left[ \left( \frac{\overline{D'_B}^{\text{lnf}} - \overline{D'_B}^{\text{vnf}}/He}{\alpha_{\text{lnf}} + (1 - \alpha_{\text{lnf}})/He} \right) T \varphi \nabla \alpha_{\text{lnf}} \right]. \quad (4.30)$$

(c) Addressing the third component on the right-hand side of equation 4.21, it is initially expressed as:

$$\nabla \cdot \left[ \left( \alpha_{\text{lnf}} \overline{D'_B}^{\text{lnf}} \overline{\phi}^{\text{lnf}} \nabla \overline{T}^{\text{lnf}} + \alpha_{\text{vnf}} \overline{D'_B}^{\text{vnf}} \overline{\phi}^{\text{vnf}} \nabla \overline{T}^{\text{vnf}} \right) \right]. \quad (4.31)$$

In cells that are entirely filled with either nanofluid or vapor,  $\nabla \overline{T}^{\text{lnf}}$  and  $\nabla \overline{T}^{\text{vnf}}$  can be replaced with  $\nabla T$ . This substitution, however, is not universally applicable at the interface. In the specific case of film boiling on a vertical surface, as investigated in this study, the interface between vapor and nanofluid typically forms an almost linear parallel alignment with the wall. Under this scenario, it is practical to consider that  $\nabla \overline{T}^{\text{lnf}}$  and  $\nabla \overline{T}^{\text{vnf}}$  are both similar in direction and magnitude to  $\nabla T$ . Furthermore, the application of Henry's equation 4.31 can be written as:

$$\nabla \cdot \left[ \left( \alpha_{\text{lnf}} \overline{D'_B}^{\text{lnf}} + \frac{\alpha_{\text{vnf}} \overline{D'_B}^{\text{vnf}}}{He} \right) \overline{\phi}^{\text{lnf}} \nabla T \right]. \quad (4.32)$$

An extended application of Henry's law leads to the formulation of the term in a singular concentration format  $\varphi$ :

$$\nabla \cdot \left[ \left( \frac{\alpha_{\text{lnf}} \overline{D'_B}^{\text{lnf}} + (1 - \alpha_{\text{lnf}}) \overline{D'_B}^{\text{vnf}}/He}{\alpha_{\text{lnf}} + (1 - \alpha_{\text{lnf}})/He} \right) \varphi \nabla T \right]. \quad (4.33)$$

There is another term called (d) in equation 4.21 which includes  $\nabla \overline{D'_B}^{\text{k}}$ . Such a term is neglected in the study by [6], and here it is neglected too.

**Analysis of the Thermophoresis Term:** The focus now shifts to the second term on the right-hand side of equation 4.18, which needs to be reinterpreted to reflect the overall volumetric concentration of nanofluid, denoted by  $\varphi$ .

## 4.2. Mathematical Framework and Governing Equations

$$\nabla \cdot \left[ \left( \alpha_{\text{lnf}} \overline{D'_{\text{Th}}}^{\text{lnf}} \overline{\phi}^{\text{lnf}} + \alpha_{\text{vnf}} \overline{D'_{\text{Th}}}^{\text{vnf}} \overline{\phi}^{\text{vnf}} \right) \nabla T \right]. \quad (4.34)$$

The refinement of this term at the fluid interface, using Henry's law, is represented by:

$$\nabla \cdot \left[ \left( \alpha_{\text{lnf}} \overline{D'_{\text{Th}}}^{\text{lnf}} + \frac{\alpha_{\text{vnf}} \overline{D'_{\text{Th}}}^{\text{vnf}}}{He} \right) \overline{\phi}^{\text{lnf}} \nabla T \right]. \quad (4.35)$$

Further employing Henry's law, the term is expressed in a singular concentration format,  $\varphi$ :

$$\nabla \cdot \left[ \left( \frac{\alpha_{\text{lnf}} \overline{D'_{\text{Th}}}^{\text{lnf}} + (1 - \alpha_{\text{lnf}}) \overline{D'_{\text{Th}}}^{\text{vnf}} / He}{\alpha_{\text{lnf}} + (1 - \alpha_{\text{lnf}}) / He} \right) \varphi \nabla T \right]. \quad (4.36)$$

The comprehensive governing equation that integrates various transport mechanisms and accounts for interfacial dynamics is then formulated as:

$$\begin{aligned} \frac{\partial \varphi}{\partial t} + \nabla \cdot (\varphi \mathbf{U}) &= \nabla \cdot (D'_B T \nabla \varphi) + \nabla \cdot (\varphi T \nabla D'_B) + \nabla \cdot (\varphi D'_B \nabla T) \\ &\quad - \nabla \cdot \left[ \left( \frac{\overline{D'_B}^{\text{lnf}} - \overline{D'_B}^{\text{vnf}} / He}{\alpha_{\text{lnf}} + (1 - \alpha_{\text{lnf}}) / He} \right) \varphi T \nabla \alpha_{\text{lnf}} \right] \\ &\quad - \nabla \cdot \left[ \left( \frac{\alpha_{\text{lnf}} \overline{D'_B}^{\text{lnf}} + (1 - \alpha_{\text{lnf}}) \overline{D'_B}^{\text{vnf}} / He}{\alpha_{\text{lnf}} + (1 - \alpha_{\text{lnf}}) / He} \right) \varphi \nabla T \right] \\ &\quad + \nabla \cdot \left[ \left( \frac{\alpha_{\text{lnf}} \overline{D'_{\text{Th}}}^{\text{lnf}} + (1 - \alpha_{\text{lnf}}) \overline{D'_{\text{Th}}}^{\text{vnf}} / He}{\alpha_{\text{lnf}} + (1 - \alpha_{\text{lnf}}) / He} \right) \varphi \nabla T \right]. \end{aligned} \quad (4.37)$$

Equation 4.37 is applicable across the entire computational domain, encompassing both the liquid nanofluid and vapor phases, as well as their interface. It effectively captures the standard transport of nanoparticles and special interface conditions, particularly where the properties of the two phases are markedly distinct.

### 4.2.4 Energy Conservation Equation

The basic form of the thermal energy conservation equation, excluding the effects of Brownian motion and thermophoresis, is as follows:

$$\frac{\partial (\rho c_p T)}{\partial t} + \nabla \cdot (\rho c_p \mathbf{U} T) = \nabla \cdot (\lambda \nabla T) - \dot{m}''' L_{fv}, \quad (4.38)$$

To incorporate the Brownian and thermophoresis contributions in the energy equation [17], the following terms are added to the right-hand side of equation 4.38 [15, 47]:

$$+\rho_{np}c_{p,np} \left[ D_B \nabla \phi \cdot \nabla T + \frac{D_{Th}}{T_\infty} \nabla T \cdot \nabla T \right] \quad (4.39)$$

To extend this equation for the entire domain, covering both nanofluid and vapor phases, it is reformulated using  $\phi$ , the volumetric concentration for the entire domain. This reformulation, as detailed in the preceding section (Section 4.2.3), results in the comprehensive energy equation:

$$\begin{aligned} \frac{\partial(\rho c_p T)}{\partial t} + \nabla \cdot (\rho c_p \mathbf{U} T) &= \nabla \cdot (k \nabla T) - \dot{m}''' L_{fv} \\ &+ \rho_{np} c_{p,np} \left( \frac{\alpha_{\text{Inf}} \overline{D'_B}^{\text{Inf}} + (1 - \alpha_{\text{Inf}}) \overline{D'_B}^{\text{vnf}} / He}{\alpha_{\text{Inf}} + (1 - \alpha_{\text{Inf}}) / He} \right) T \nabla \phi \nabla T \\ &+ \rho_{np} c_{p,np} \left( \frac{\alpha_{\text{Inf}} \overline{D'_{Th}}^{\text{Inf}} + (1 - \alpha_{\text{Inf}}) \overline{D'_{Th}}^{\text{vnf}} / He}{\alpha_{\text{Inf}} + (1 - \alpha_{\text{Inf}}) / He} \right) \phi \nabla T \nabla T. \end{aligned} \quad (4.40)$$

This research employs OpenFOAM, an open-source C++ CFD toolkit [61], as its computational foundation. Accurate phase change modeling is crucial for determining the mass flux rate  $\dot{m}'''$ . For this purpose, the interfaceHeatResistance model from OpenFOAM v2006 is used. The model is based on principles outlined by Hardt and Wondra [62].

## 4.2.5 Updating Thermo-Physical Properties Of Nanofluid

The thermophysical properties of the nanofluid, such as density, specific heat, viscosity, thermal conductivity, and surface tension coefficient, are recalculated at each time step following the update in the volumetric fraction of nanoparticles.

**Estimation of Density and Specific Heat Capacity** To calculate the density and specific heat of the nanofluid, the study employs the following relevant formulas [16, 47, 63]:

$$\rho_{nf} = \phi \rho_{np} + (1 - \phi) \rho_{bf}, \quad (4.41)$$

$$c_{p,nf} = \frac{\phi \rho_{np} c_{p,np} + (1 - \phi) \rho_{bf} c_{p,bf}}{\rho_{nf}}, \quad (4.42)$$

## 4.2. Mathematical Framework and Governing Equations

In these equations,  $\varphi$  represents the volume fraction of nanoparticles, with 'np', 'bf', and 'nf' referring to nanoparticles, base fluid, and nanofluid, respectively.

**Estimation of Thermal Conductivity and Dynamic Viscosity in Nanofluids** To evaluate the thermal conductivity and dynamic viscosity of nanofluids, this study applies established correlations found in the literature, notably those in [1]. The formulas used for these properties are:

$$\lambda_{\text{nf}} = \lambda_{\text{bf}} \frac{\lambda_{\text{np}} + 2\lambda_{\text{bf}} + 2\varphi(\lambda_{\text{np}} - \lambda_{\text{bf}})}{\lambda_{\text{np}} + 2\lambda_{\text{bf}} - \varphi(\lambda_{\text{np}} - \lambda_{\text{bf}})}, \quad (4.43)$$

$$\mu_{\text{nf}} = \frac{\mu_{\text{bf}}}{(1 - \varphi)^{2.5}}. \quad (4.44)$$

In these expressions,  $\lambda_{\text{nf}}$  and  $\mu_{\text{nf}}$  indicate the nanofluid's thermal conductivity and dynamic viscosity, respectively.  $\lambda_{\text{np}}$  is the nanoparticles' thermal conductivity, while  $\lambda_{\text{bf}}$  and  $\mu_{\text{bf}}$  correspond to the base fluid's thermal conductivity and dynamic viscosity.  $\varphi$  is the volume fraction of nanoparticles.

**Modeling Surface Tension** The surface tension as a function of nanoparticle concentration is determined using a model proposed by Szyszkowski [64]. The relationship is expressed by:

$$\frac{\sigma_{\text{bf}} - \sigma_{\text{nf}}}{\sigma_{\text{bf}}} = b \ln \left( \frac{\varphi}{a} + 1 \right), \quad (4.45)$$

Empirical values for the coefficients  $a$  and  $b$  are set at  $a = 7.673 \times 10^7$  and  $b = 7.773 \times 10^3$  [65].

### 4.2.6 Validation With Taylor 2D Film Boiling Simulations

The solver for the governing equations has been validated by comparing its results in Taylor 2D film boiling simulations for a pure fluid with well-known benchmarks. Notably, the Taylor film boiling simulation is a standard for horizontal film boiling, whereas the main focus of the study is on vertical film boiling. The choice of the Taylor benchmark for validation is grounded on its established credibility, the presence of empirical correlations for comparative analysis, and its relevance under near-critical conditions, which are similar to those encountered in the vertical film boiling investigations. The essential thermophysical properties are shown in Table 4.1. These simulations are conducted for the base fluid ( $\varphi = 0$ ). The boundary and initial conditions of the benchmark case, along with the geometry of the computational domain, are illustrated in Fig. 4.1. For more details regarding the geometry and initial conditions of this benchmark, readers are directed to chapters 2 and 3 and the 2D film boiling benchmark case.

**Table 4.1:** Fluid characteristics of saturation water at 21.9 MPa and 346 K employed to address the benchmark case of 2D film boiling and 2D axisymmetric vertical film boiling.

	<b>Dimension</b>	<b>Base Fluid</b>	<b>Vapor</b>
Thermal conductivity, $\lambda$	$\text{W m}^2 \text{K}^{-1}$	0.545	0.54
Density, $\rho$	$\text{kg m}^{-3}$	402.4	242.7
Viscosity, $\mu$	$\text{Pa s}$	$4.53 \times 10^{-5}$	$3.23 \times 10^{-5}$
Specific heat capacity $c_p$	$\text{J kg}^{-1} \text{K}^{-1}$	$2.18 \times 10^5$	$3.52 \times 10^5$
Latent heat, $h$	$\text{J kg}^{-1}$	$2.24 \times 10^6$	$1.96 \times 10^6$
Surface tension, $\sigma$	$\text{N m}^{-1}$	$7 \times 10^{-5}$	

The computation of the Nusselt number follows the method outlined by [66]:

$$\text{Nu} = \frac{\int_0^L \left( \frac{L_{\text{char}}}{\Delta T} \frac{\partial T}{\partial y} \Big|_{y=0} \right) dx}{L}, \quad (4.46)$$

where  $L_{\text{char}}$  is defined as:

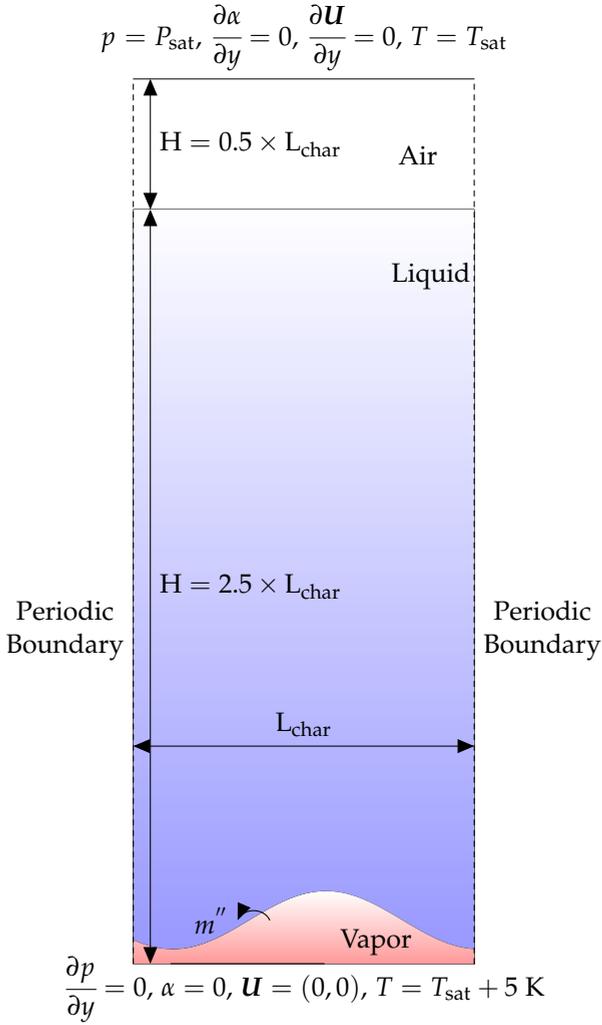
$$L_{\text{char}} = \sqrt{\frac{\sigma}{(\rho_l - \rho_v)g}}. \quad (4.47)$$

The calculated Nusselt number from the numerical study is then compared with the Nusselt number from Berenson's experimental correlation [67]. The 2D film boiling Nusselt number according to Berenson is given by:

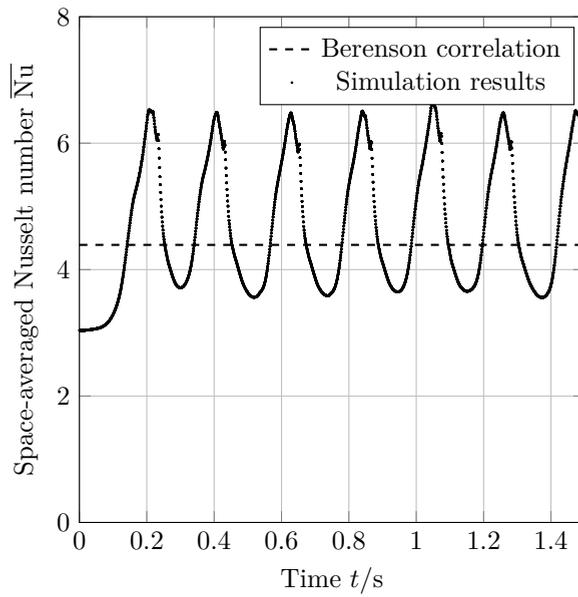
$$\text{Nu} = 0.425 \left[ \frac{\rho_v(\rho_l - \rho_v)g(h_v - h_l)}{\lambda_v \mu_v \Delta T} \right]. \quad (4.48)$$

As depicted in Fig. 4.2, the numerical simulation yields an average surface Nusselt number of 4.6, while Berenson's formula estimates it at 4.25, showing a 7.6% variance. This comparative analysis confirms the alignment of the numerical simulation results with Berenson's empirical model.

## 4.2. Mathematical Framework and Governing Equations



**Fig. 4.1:** Boundary and initial conditions along with computational domain for the 2D film boiling simulations.



**Fig. 4.2:** Comparison between Nusselt numbers from numerical simulations and Berenson's experimental findings in 2D film boiling of pure water.

### 4.3 Problem Description

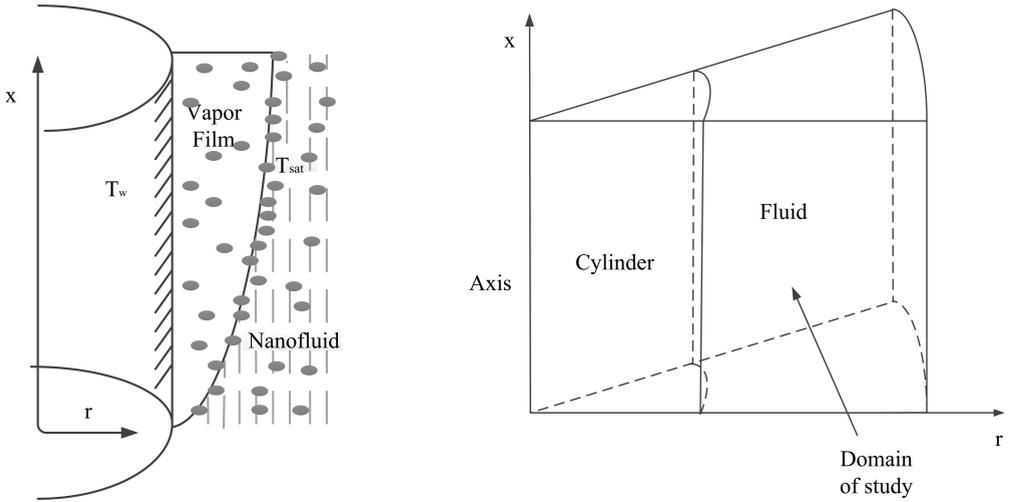
As shown in Fig. 4.3, this research studies a vertical cylinder immersed in a nanofluid. The simulation initiates with a thin vapor layer ( $\delta = 0.01$  mm) attached to the cylinder's surface. When the temperature of the cylinder wall,  $T_w$ , surpasses the saturation temperature  $T_{\text{sat}}$ , the thickness of the attached vapor film grows along the cylinder's radial axis. It is important to note that while nanofluid boiling involves phenomena such as varying triple-line motion due to nanoparticle deposition, this study focuses primarily on film boiling. In film boiling conditions, the presence of an insulating vapor film significantly mitigates these effects by minimizing the direct contact between the liquid and the surface. As a result, the impact of changes in the triple-line angle is considerably reduced and is therefore not a consideration in this analysis.

The computational domain, depicted in Fig. 4.3b, is modeled as a two-dimensional (2D) axisymmetric representation of the actual three-dimensional (3D) setup. This domain is aligned such that the x-axis coincides with the vertical centerline of the cylinder, serving as the axis of symmetry. The domain's geometry is wedge-shaped, with its front and back boundaries established as wedge-type boundaries within the OpenFOAM framework. Employing wedge boundaries in OpenFOAM is advantageous for simulating 2D axisymmetric conditions, effectively capturing the physical processes while reducing computational demands. The cylinder in the simulation has a length of 2 mm ( $L = 2$  mm) and a radius of 0.05 mm ( $R = 0.05$  mm). At the macroscopic scale, the process is assumed to be isobaric, maintaining a constant pressure  $P_{\text{sat}}$  at the far-field boundaries, where the fluid is in thermodynamic equilibrium with the environment.

Regarding the boundaries of the computational domain, they are characterized as follows:

- The distant boundary (right side) is considered a free stream and is treated as an open boundary in the simulation. This implies that the gradients of temperature, volume fraction, and velocity across this boundary are zero, reflecting a far-field condition where the influence of the cylinder is negligible. This boundary is located at a radius of 0.4 mm ( $R = 0.4$  mm).
- The top and bottom boundaries are also open, allowing for the free inflow/outflow of vapor and fluid, which is consistent with the assumption of an unbounded domain in the vertical direction.

The boundary conditions applied to the governing equations are inspired by the studies conducted by Avramenko et al. [51, 68] and Malvandi et al. [17],



(a) Vapor layer development on the vertical cylinder.

(b) Axisymmetric computational setup for simulation.

**Fig. 4.3:** Geometrical configurations analyzed: (a) Vertical cylinder in nanofluid flow with vapor layer formation, (b) Axisymmetric computational setup for simulations.

and are designed to reflect the physical scenario of a vertical cylinder in a nanofluid as closely as possible.

The boundary conditions applied to the governing equations are specified as follows:

- At the base:

$$T = T_{sat}$$

$$P = P_{sat}$$

$$\alpha = 1$$

$$\frac{\partial \mathbf{U}}{\partial x} = 0$$

- At the distant boundary (right side), reflecting the free stream condition:

$$\frac{\partial T}{\partial r} = \frac{\partial \alpha}{\partial r} = \frac{\partial \mathbf{U}}{\partial r} = 0$$

$$P = P_{sat}$$

- At the outlet (top):

$$\frac{\partial P}{\partial x} = \frac{\partial \mathbf{U}}{\partial x} = \frac{\partial T}{\partial x} = 0$$

$$\alpha = 1$$

### 4.3. Problem Description

- Along the heated wall (left side):

$$T = T_{\text{sat}} + 5 \text{ K}$$

$$\frac{\partial P}{\partial r} = \alpha = \mathbf{U} = 0$$

$$\left( D_B \frac{d\phi}{dr} \right) = - \left( \frac{D_{\text{Th}}}{T_\infty} \frac{dT}{dr} \right)$$

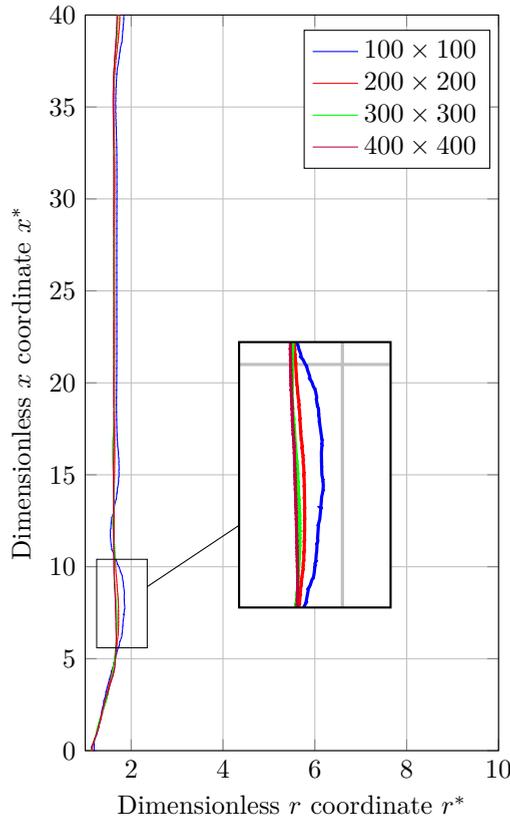
The final boundary condition stems from the principle that nanoparticles cannot permeate through the wall, implying that the net flux of nanoparticles at the wall surface must equal zero. The equation's terms represent the overall movement of nanoparticles, known as Stefan's flow, which results from differences in concentration and temperature gradients [17, 51, 68].

The properties of the base fluid, its vapor, and the nanoparticles utilized in this study are summarized in Tables 4.1 and 4.2.

**Table 4.2:** Properties of  $\text{Al}_2\text{O}_3$  nanoparticles.

Properties	Units	$\text{Al}_2\text{O}_3$
Thermal conductivity, $\lambda$	$\text{W m}^2 \text{K}^{-1}$	40
Density, $\rho$	$\text{kg m}^{-3}$	3970
Specific heat, $c_p$	$\text{J kg}^{-1} \text{K}^{-1}$	765

A mesh sensitivity analysis was carried out to determine the most suitable grid size for the upcoming stages of the research. This analysis primarily focused on the behavior of the vapor film interface at a specific time for different grid resolutions. The grid sizes evaluated were  $N_{\text{grids}} = [100 \times 100, 200 \times 200, 300 \times 300, 400 \times 400]$ . The outcomes of this investigation are presented in Fig. 4.4. Based on this assessment, the  $300 \times 300$  grid was selected for further analysis due to its close agreement with the results obtained using the  $400 \times 400$  grid. The grid was structured utilizing a simpleGrading scheme of (15, 1, 1) within the OpenFOAM environment to ensure a refined mesh near the cylinder while maintaining a coarser grid further away, optimizing the computational resources.



**Fig. 4.4:** Visualization of vapor film interfaces at a specific time for various grid sizes  $N_{\text{grids}} = [100 \times 100, 200 \times 200, 300 \times 300, 400 \times 400]$ . The  $300 \times 300$  grid was chosen for further study owing to its performance akin to the  $400 \times 400$  grid.

## 4.4 Results And Analysis

This section delves into the solution and examination of the 2D axisymmetric film boiling around a cylinder, as outlined earlier. The focus is on parameters such as nanofluid volume fraction, temperature, and nanoparticle concentration. Additionally, key thermophysical properties like density, specific heat capacity, thermal conductivity, kinematic viscosity, and surface tension are scrutinized. These factors are analyzed across different regions of the geometry, namely the vapor, liquid, and vapor-liquid interface, and are tracked over time until equilibrium is achieved. Furthermore, the study evaluates the Nusselt number under various nanoparticle concentrations, shedding light on the impact of nanoparticles on heat transfer and boiling efficiency.

#### 4.4.1 Nanofluid Volume Fraction

Fig. 4.5 depicts the temporal variation of the nanofluid volume fraction ( $\alpha$ ), showing its progression towards a steady state. The nanoparticle concentration in the liquid phase is  $5 \times 10^{-2}$ . This illustration utilizes dimensionless forms of the axial coordinate ( $r$ ) and time ( $t$ ). The equations for these dimensionless variables are as follows:

$$r^* = \frac{r}{\delta_0}, \quad (4.49)$$

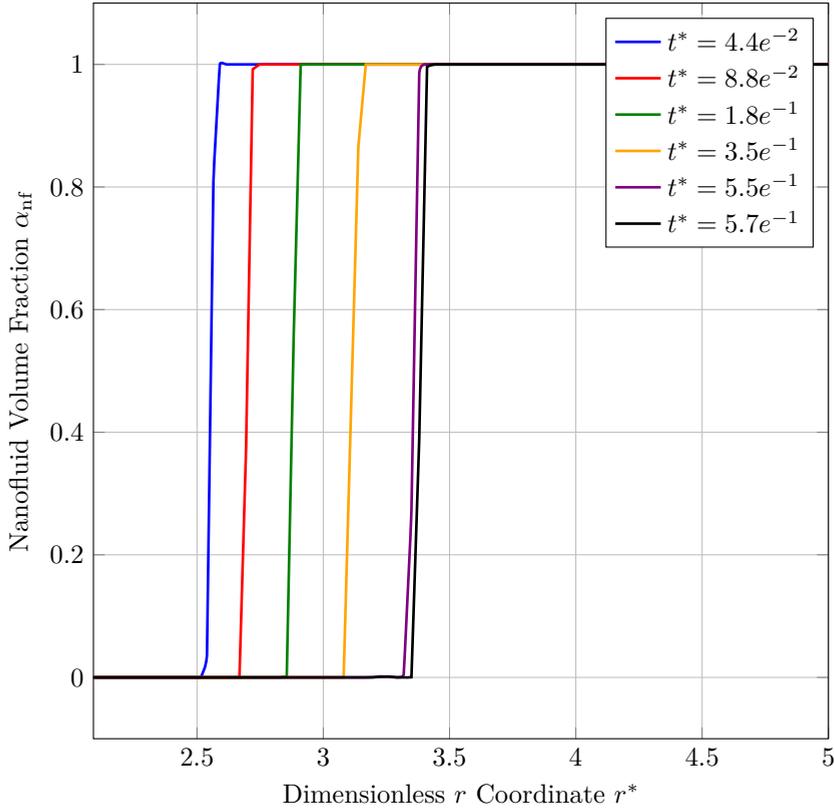
with  $\delta_0$  determined by the analytical expression [15]:

$$\delta_0 = \left( \frac{4\lambda_v \mu_v \Delta T x}{g L_{fv} \rho_f \rho_v} \right)^{\frac{1}{4}} \quad (4.50)$$

The dimensionless time variable is given by:

$$t^* = \frac{\tau}{\text{Pr}}, \quad \tau = \frac{t \mu_v}{\delta_0^2 \rho_v} \quad (4.51)$$

The graph exhibits a distinct transition, notably a rapid change from  $\alpha = 0$  (representing the vapor phase) to  $\alpha = 1$  (indicating the liquid nanofluid phase), pinpointing the vapor-liquid interface location. This boundary progressively shifts axially over time, demonstrating the vapor film's increasing thickness. This growth persists until reaching a near-equilibrium state at  $t^* = 5.5 \times 10^{-1}$ . In this equilibrium phase, the interface shows minimal movement, fluctuating gently in its location.



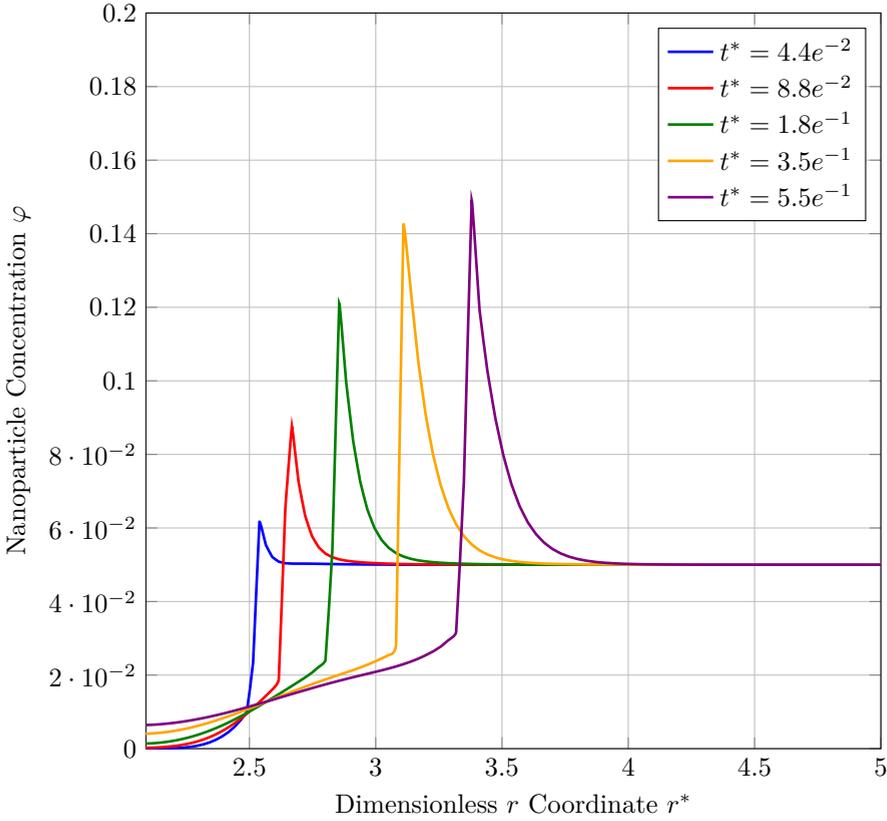
**Fig. 4.5:** Development of the nanofluid volume fraction ( $\alpha_{\text{nf}}$ ) along the dimensionless radial coordinate ( $r^*$ ) at different dimensionless times ( $t^*$ ). Observations were made using line probe techniques at a fixed axial point ( $x^* = 70$ ) on the cylinder's vertical axis.

#### 4.4.2 Nanoparticles Concentration

Fig. 4.6 illustrates the temporal evolution of nanoparticle concentration ( $\varphi$ ) along the dimensionless radial coordinate  $r^*$ . Initially, the nanoparticle concentration in the liquid phase is  $5 \times 10^{-2}$ . At the gas-liquid interface, it is observed that a significant rise in nanoparticle concentration, surpassing the initial values. This increment is consistent with the theoretical understanding of the enrichment of nanoparticles at the liquid-vapor interface during nanofluid evaporation that nanoparticles tend to migrate toward the interface during evaporation, a concept supported by multiple studies including those referenced by Stephan et al. [69, 70] and Yong et al. [71].

Notably, the higher concentration observed near the interface diminishes along the axial direction, ultimately converging to the original nanoparticle concentration in the nanofluid, indicative of a system reaching equilibrium.

#### 4.4. Results and Analysis



**Fig. 4.6:** Dynamic axial change in nanoparticle concentration ( $\varphi$ ) along the dimensionless radial coordinate ( $r^*$ ) over various dimensionless time frames ( $t^*$ ), culminating in a steady state. The concentration enhancement at the interface underpins the theoretical phenomena observed in prior research. Measurements were conducted using line probes at a consistent axial location ( $x^* = 70$ ) on the cylinder's vertical axis.

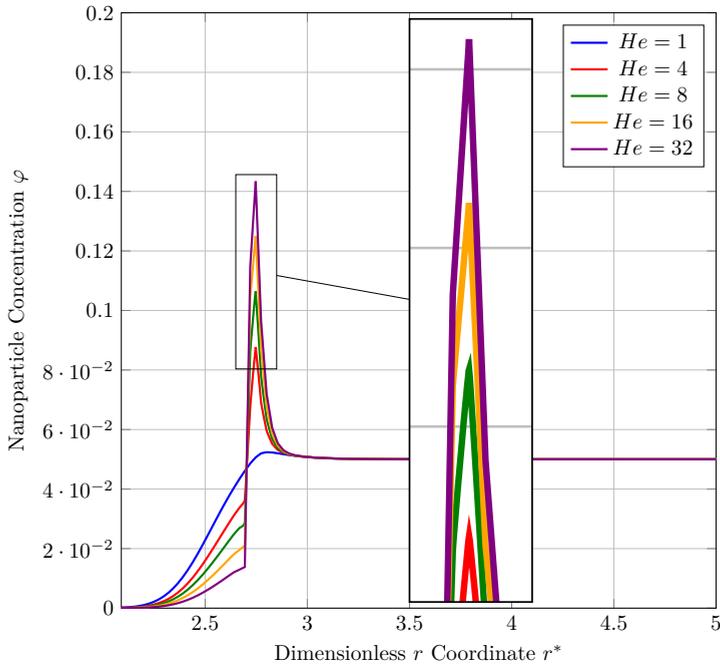
The domain's right boundary is considered open, allowing for the free flow of the fluid, thus reinforcing the validity of the observed concentration profiles.

The observed concentration enrichment at the interface is indicative of the complex interfacial behavior that is characteristic of nanofluid evaporation. The CFD model deployed in this study successfully captures this behavior and, to the best of the authors' knowledge, may be the first to provide such detailed predictive insights.

The variable temporal behaviors and the concentration distributions, which transition from a non-linear profile to a linear one as a steady state is approached, are in alignment with analytical models such as those presented by Avramenko et al. [15] and Malvandi et al. [17].

As mentioned in equation 4.16, the model for considering nanoparticles

governing equation in this study, incorporates specific conditions, commonly known as "jump conditions," which are pivotal in predicting the enrichment of nanoparticles at the interface. The solubility disparity of nanoparticles between the vapor and liquid phases is addressed by applying Henry's law, with  $He$  being the distribution coefficient that adjusts for the differential solubility.



**Fig. 4.7:** Impact of the Henry's law coefficient ( $He$ ) on the nanoparticle concentration profiles at the liquid-vapor interface. Each curve represents a different  $He$  value, illustrating how the distribution coefficient affects the enrichment of nanoparticles.

Fig. 4.7 illustrates the impact of different  $He$  values on the nanoparticle concentration profiles, with a particular emphasis on the interface. The enrichment of nanoparticles at the interface is well-documented in the scientific literature [69–71], and this CFD model's capability to predict such behavior is a novel contribution to the field. The figure indicates that higher  $He$  values lead to an increased concentration of nanoparticles at the interface. Furthermore,  $He = 1$  represents the scenario where nanoparticle solubility is the same in both the nanofluid and its vapor, an unlikely situation in practice. This specific curve aligns with the concentration profiles obtained by Malvandi et al. [17].

The value of  $He = 8$  is assumed for all results presented in this chapter. Determining the exact  $He$  value is outside the scope of this study, as it re-

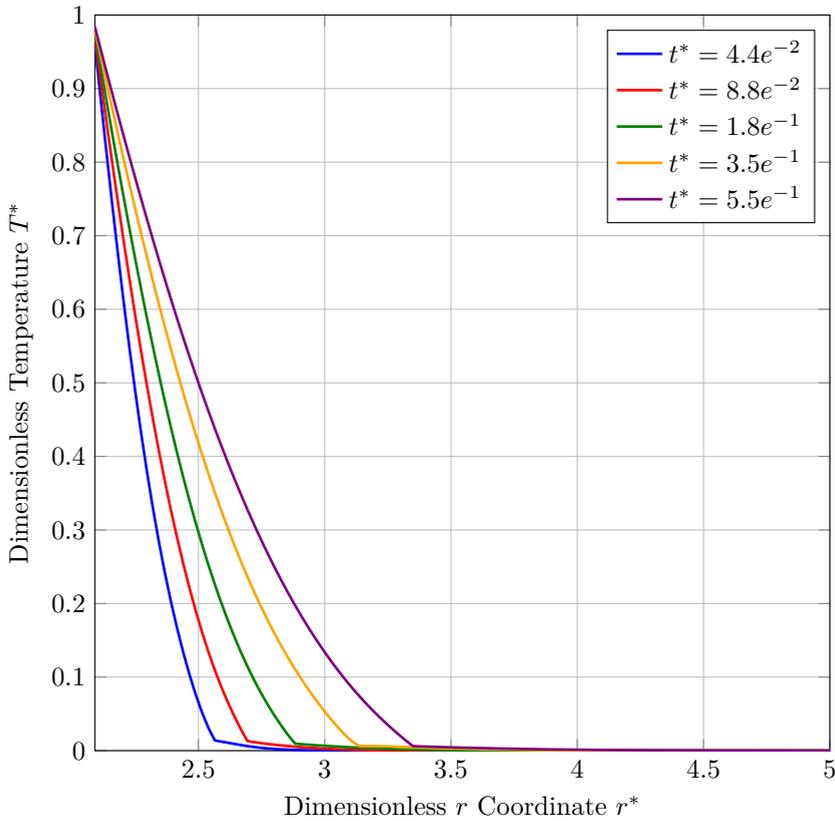
quires experimental work to quantify the degree of nanoparticle enrichment at the liquid-vapor interface and subsequently calculate the appropriate  $He$  values. However, it is important to note that this study proposes a mathematical CFD model capable of predicting such enrichment. The conclusion chapter will highlight the necessity of experimental validation to determine the precise  $He$  values, which would enhance the verification of the interfacial enrichment concentrations predicted by the model.

### 4.4.3 Temperature Distribution

The dimensionless temperature in the study is defined as follows:

$$T^* = \frac{T - T_{\text{sat}}}{T_{\text{wall}} - T_{\text{sat}}}, \quad (4.52)$$

In Fig. 4.8, the variation in dimensionless temperature within the vapor area is depicted. The boundaries for vapor, interface, and liquid domains can be identified using Fig. 4.5. From the wall towards the nanofluid liquid domain, the temperature shows a decline, settling just above the saturation temperature. Near the wall, a sharp linear decrease in temperature is observed, which gradually lessens as it approaches the vapor-liquid interface. The profiles demonstrate a nearly linear decrease in temperature from the heated wall towards the bulk of the liquid, with a subtle curvature in the middle region resembling the characteristics of temperature profile for nanofluid boiling seen in Fig. 6 of Malvandi et al. [47] or Fig. 2 of Avramenko et al [4].



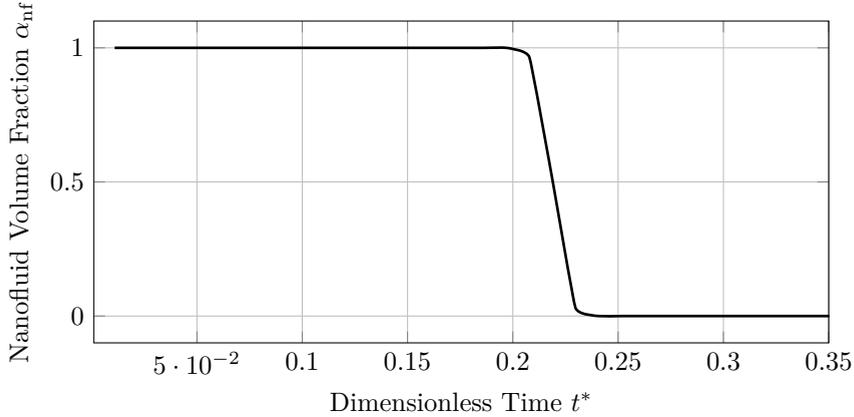
**Fig. 4.8:** Changes in the dimensionless temperature ( $T^*$ ) along the radial coordinate ( $r^*$ ) over different dimensionless time periods ( $t^*$ ). Data were gathered using line probe methods at a set axial location ( $x^* = 70$ ) on the cylinder's vertical axis.

#### 4.4.4 Nanofluid Volume Fraction And Nanoparticle Concentration Over Time

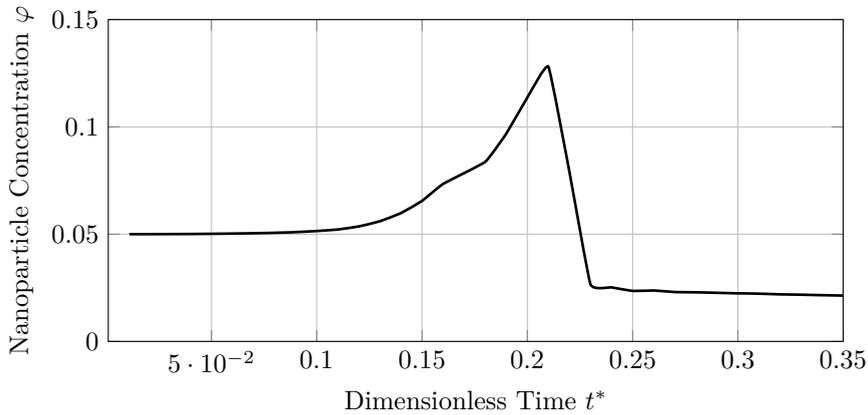
Fig. 4.9 illustrates the changes in nanofluid volume fraction and nanoparticle concentration at a location near the wall ( $r^* = 3, x^* = 70$ ) over time. Initially, this location is submerged in the liquid phase of the nanofluid, as indicated by the volume fraction being 1. The nanoparticle concentration begins at 0.05, which is the studied initial nanoparticle concentration. Over time, there is an increase in nanoparticle concentration at this point, although it remains within the liquid domain. This increase suggests the gas-liquid interface is approaching. The figure captures the moment when the interface reaches this point at a dimensionless time of approximately 1, where the highest concentration of nanoparticles is recorded. As the interface passes and the point enters the vapor phase (volume fraction falls to 0), there is a marked

#### 4.4. Results and Analysis

reduction in nanoparticle concentration, dropping below the initial value. Following this, the nanoparticle concentration experiences a gradual decline, suggesting the dispersion of nanoparticles in the vapor phase.



(a)



(b)

**Fig. 4.9:** Dynamics of (a) nanofluid volume fraction ( $\alpha_{nf}$ ) and (b) nanoparticle concentration ( $\varphi$ ) at a wall-near probe ( $r^* = 3, x^* = 70$ ) over dimensionless time.

#### 4.4.5 Nanofluid Thermophysical Characteristics

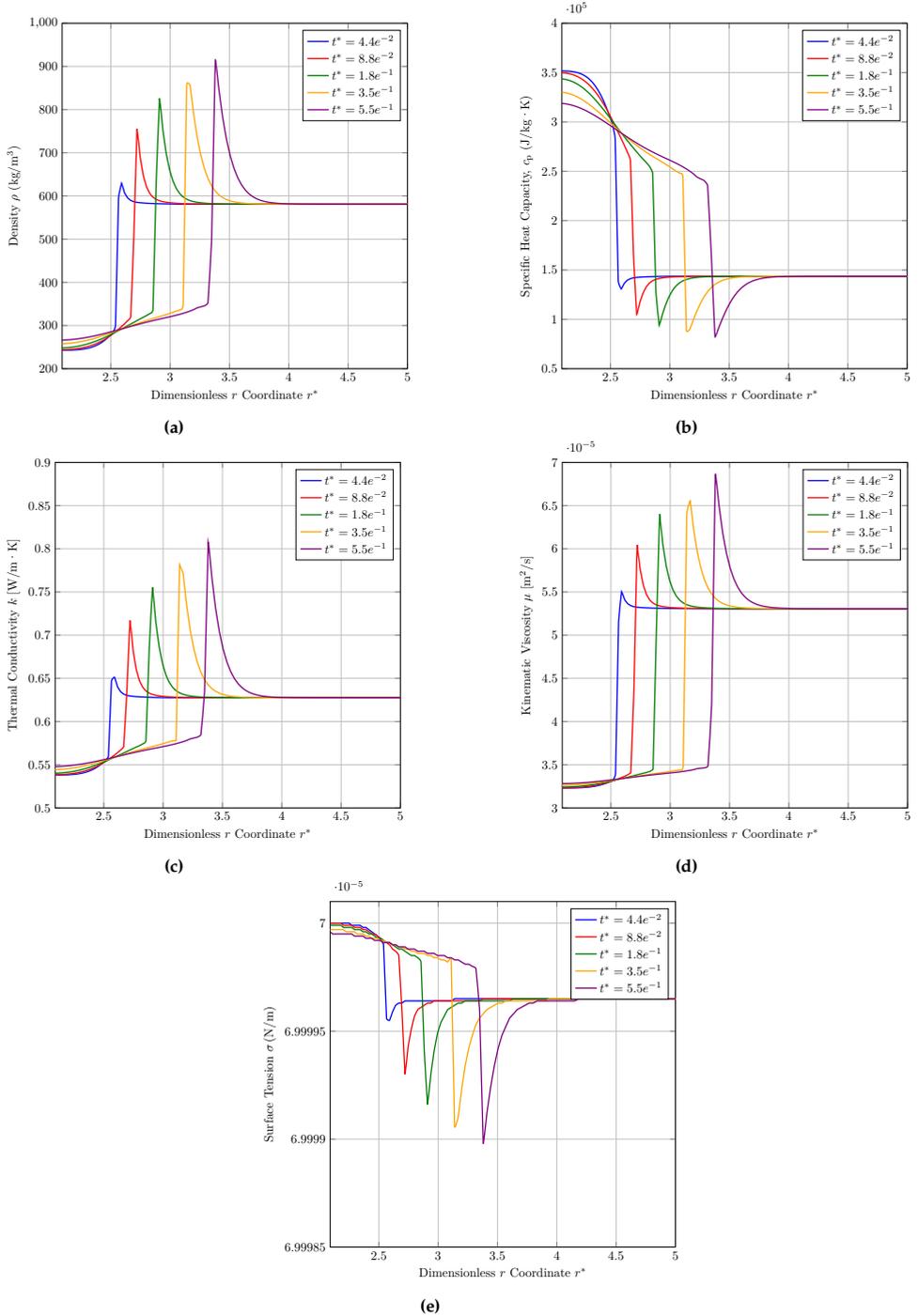
Fig. 4.10 illustrates the temporal evolution of several key thermophysical properties of a nanofluid during film boiling. These properties, tracked over various dimensionless time intervals ( $t^*$ ), include density ( $\rho$ ), specific heat capacity ( $c_p$ ), thermal conductivity ( $k$ ), kinematic viscosity ( $\mu$ ), and surface tension ( $\sigma$ ).

This data, when correlated with the nanoparticle concentration trends shown in Fig. 4.6, reveals a pronounced correlation between nanoparticle concentration in the fluid and its physical properties. It is noteworthy that the changes in these properties are quantified as percentages, based on the extreme values at the interface during steady-state conditions compared to the baseline properties of the liquid nanofluid:

- The density ( $\rho$ ) shows an approximate 43% increase, as seen in Fig. 4.10a.
- Specific heat capacity ( $c_p$ ) decreases by about 36.03%, as shown in Fig. 4.10b.
- Thermal conductivity ( $k$ ) experiences an increase near 29%, evident in Fig. 4.10c.
- Kinematic viscosity ( $\mu$ ) undergoes an increase close to 30%, as depicted in Fig. 4.10d.
- Surface tension ( $\sigma$ ) exhibits a marginal decrease, roughly 0.00064%, maintaining relative stability compared to other properties, as observable in Fig. 4.10e.

These trends indicate a direct relationship between the presence of nanoparticles and increases in density, thermal conductivity, and kinematic viscosity. On the other hand, specific heat capacity and surface tension show a decrease with higher nanoparticle concentrations. These findings highlight the significant impact of nanoparticle concentration on the essential characteristics and behavior of nanofluids, underlining the importance of accurate nanoparticle measurement in fluid dynamics and CFD studies.

#### 4.4. Results and Analysis



**Fig. 4.10:** Axial variations of thermophysical properties in a transient state, tracked until equilibrium is reached at a constant axial point ( $x^* = 70$ ) along the cylinder. The subfigures display the changes over time in (a) density ( $\rho$ ), (b) specific heat capacity ( $c_p$ ), (c) thermal conductivity ( $k$ ), (d) kinematic viscosity ( $\nu$ ), and (e) surface tension ( $\sigma$ ).

#### 4.4.6 Nusselt Number

The calculation of the normalized Nusselt number for film boiling on a vertical plate is based on the methodology provided by [4]:

$$\overline{\text{Nu}} = \frac{K(\varphi)}{\sqrt{\frac{\mu_v c_p}{\text{Pr} k_v \delta_0^2}}} \left. \frac{\partial T}{\partial r} \right|_{r=0'} \quad (4.53)$$

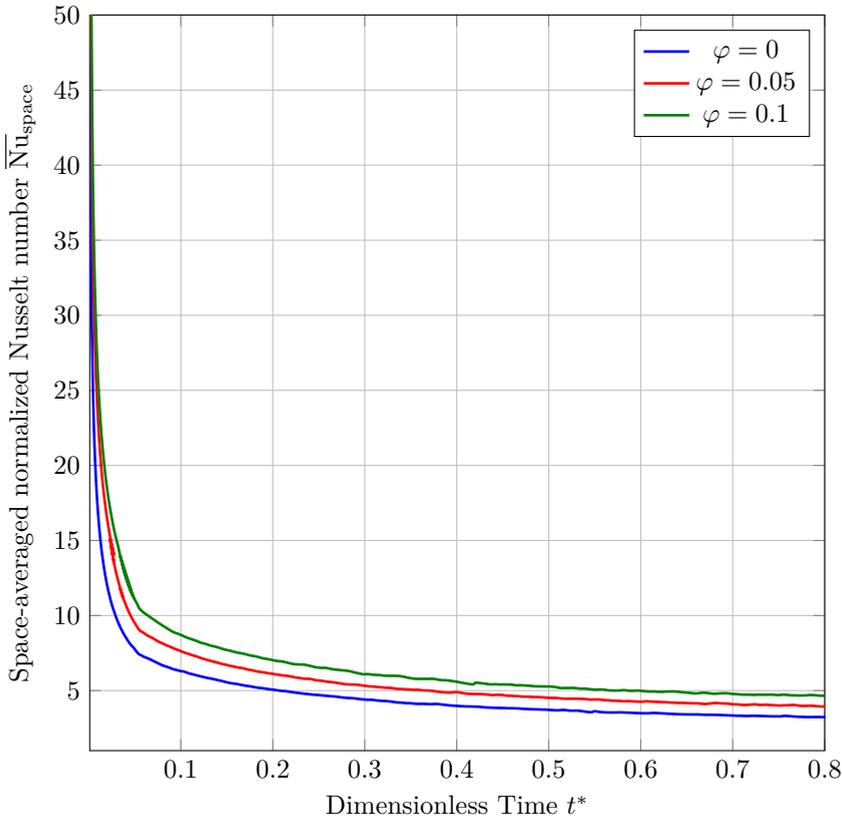
where  $K(\varphi)$  represents the thermal conductivity of the vapor nanofluid at the heated wall's surface. In this research, the focus is on the space-averaged normalized Nusselt number ( $\overline{\text{Nu}}_{\text{space}}$ ), which is defined as:

$$\overline{\text{Nu}}_{\text{space}} = \frac{\int_0^L \left( \frac{K(\varphi)}{\sqrt{\frac{\mu_v c_p}{\text{Pr} k_v \delta_0^2}}} \left. \frac{\partial T}{\partial r} \right|_{r=R} \right) dr}{L} \quad (4.54)$$

Fig. 4.11 illustrates the changes in the space-averaged normalized Nusselt number, an indicator of heat transfer efficiency, over dimensionless time  $t^*$ . Initially, at the start of the heating process, the Nusselt number is notably high. As the process continues, the Nusselt number decreases, eventually reaching a stable state, indicative of a consistent heat transfer rate.

The addition of nanoparticles to the fluid, denoted as  $\varphi$ , significantly influences the heat transfer efficiency. When the nanoparticle concentration is  $\varphi = 0.05$ , there is an increase of approximately 21.93% in the Nusselt number compared to the base fluid, where  $\varphi = 0$ . At a concentration of  $\varphi = 0.1$ , the Nusselt number escalates by 43.84%. These findings underscore the improved heat transfer capabilities with the increasing concentration of nanoparticles in the fluid.

## 4.5. Conclusion



**Fig. 4.11:** Variation of the space-averaged normalized Nusselt number,  $\overline{Nu}_{space}$ , over dimensionless time  $t^*$ , indicating heat transfer trends.

## 4.5 Conclusion

This chapter has addressed critical limitations within existing CFD research on nanofluid boiling. The literature review reveals two main areas where current theoretical CFD models fall short: first, most CFD studies do not include a detailed governing equation for nanoparticle concentration. When such equations do exist, they typically apply only to the vapor phase, ignoring the complex behaviors in the liquid nanofluid phase and at the vapor-liquid interface. Second, the absence of a comprehensive model hinders the in-depth analysis of nanoparticle concentration dynamics specifically at the vapor-liquid interface. This gap significantly limits the capacity of CFD models to predict and simulate the phenomenon of nanoparticle enrichment at this critical juncture.

In addressing the challenges outlined, this thesis presents an innovative

approach by integrating a well-established governing equation for nanoparticle concentration with the Continuous-Species-Transfer (CST) method, previously introduced by Marschall et al. [6], into the Computational Multi-Fluid Dynamics (CMFD) framework. Traditionally applied to single-phase scenarios, this governing equation is now expanded to encompass the complexities of both vapor and liquid phases, as well as the interface between them. The applicability of this method was shown through a 2D axisymmetric film boiling analysis on a vertical cylinder.

The axisymmetric film boiling study examined aspects such as nanoparticle concentration, temperature, and nanofluid volume fraction. The results demonstrated evolving transient behaviors leading to a stable state, including an increase in concentration at the gas-liquid interface due to vapor film evaporation. This research is the first to predict such behavior using a CFD model.

Changes in the thermophysical properties of nanofluids during film boiling were significant compared to their baseline characteristics under steady-state conditions. These included a 43% increase in density, a 36.03% decrease in specific heat capacity, a 29% rise in thermal conductivity, and a 30% increase in kinematic viscosity, measured across various dimensionless times. Surface tension experienced a slight decrease of about 0.00064%.

This study also evaluated the heat transfer efficiency in nanofluid boiling. The space-averaged Nusselt number, serving as an indicator of heat transfer efficiency, was found to stabilize over time, indicating a consistent heat transfer rate. Notably, the nanoparticle concentration significantly influenced the heat transfer process. For instance, a nanoparticle concentration of  $\varphi = 0.05$  resulted in a Nusselt number increase of approximately 21.93% compared to the base fluid. At a higher concentration of  $\varphi = 0.1$ , the increase in the Nusselt number reached 43.84%. These results underscore that greater concentrations of nanoparticles in the fluid improve heat transfer efficiency, as reflected by the increased Nusselt numbers.

## References

- [1] J. Buongiorno, "Convective transport in nanofluids," 2006.
- [2] C. Yang, W. Li, and A. Nakayama, "Convective heat transfer of nanofluids in a concentric annulus," *International Journal of Thermal Sciences*, vol. 71, pp. 249–257, 2013.
- [3] A. Malvandi and D. D. Ganji, "Effects of nanoparticle migration on force convection of alumina/water nanofluid in a cooled parallel-plate channel," *Advanced Powder Technology*, vol. 25, no. 4, pp. 1369–1375, 2014.
- [4] A. Avramenko, I. V. Shevchuk, N. P. Dmitrenko, A. A. Moskalenko, and P. N. Logvinenko, "Unsteady convective heat transfer in nanofluids at instantaneous transition to film boiling," *International Journal of Thermal Sciences*, vol. 164, p. 106873, 2021.
- [5] Y. Lin and Y. Jiang, "Effects of Brownian motion and thermophoresis on nanofluids in a rotating circular groove: A numerical simulation," *International Journal of Heat and Mass Transfer*, vol. 123, pp. 569–582, 8 2018.
- [6] H. Marschall, K. Hinterberger, C. Schüler, F. Habla, and O. Hinrichsen, "Numerical simulation of species transfer across fluid interfaces in free-surface flows using OpenFOAM," *Chemical Engineering Science*, vol. 78, pp. 111–127, 8 2012.
- [7] F. Razmmand, R. Mehdipour, and S. M. Mousavi, "A numerical investigation on the effect of nanofluids on heat transfer of the solar parabolic trough collectors," *Applied Thermal Engineering*, vol. 152, pp. 624–633, 2019.
- [8] S. Azizifar, M. Ameri, and I. Behroyan, "Experimental investigation of the subcooled flow boiling heat transfer of water and nanofluids in a horizontal metal foam tube." [Online]. Available: <https://doi.org/10.1007/s00231-021-03042-9>
- [9] E. Raki, M. Afrand, and A. Abdollahi, "Influence of magnetic field on boiling heat transfer coefficient of a magnetic nanofluid consisting of cobalt oxide and deionized water in nucleate regime: An experimental study," *International Journal of Heat and Mass Transfer*, vol. 165, p. 120669, 2021.
- [10] M. M. Sarafray and A. T. K. Abad, "Statistical and experimental investigation on flow boiling heat transfer to carbon nanotube-therminol nanofluid," *Physica A: Statistical Mechanics and its Applications*, vol. 536, p. 122505, 2019.
- [11] E. Akbari, S. G. Holagh, H. Saffari, and M. Shafiee, "On the effect of silver nanoparticles deposition on porous copper foams on pool boiling heat transfer enhancement: an experimental visualization," *Heat and Mass Transfer/Waerme- und Stoffuebertragung*, vol. 58, no. 3, pp. 447–466, 3 2022.
- [12] M. A. Green, "The integration of liquid cryogen cooling and cryocoolers with superconducting electronic systems," *Superconductor Science and Technology*, vol. 16, no. 12, p. 1349, 2003.
- [13] S. A. Ebrahim, S. Chang, F. B. Cheung, and S. M. Bajorek, "Parametric investigation of film boiling heat transfer on the quenching of vertical rods in water pool," *Applied Thermal Engineering*, vol. 140, pp. 139–146, 7 2018.

## References

- [14] J. Chen, R. Zeng, X. Zhang, L. Qiu, and J. Xie, "Numerical modeling of flow film boiling in cryogenic chilldown process using the AIAD framework," *International Journal of Heat and Mass Transfer*, vol. 124, pp. 269–278, 2018.
- [15] A. A. Avramenko, I. V. Shevchuk, A. I. Tyrinov, and D. G. Blinov, "Heat transfer at film condensation of moving vapor with nanoparticles over a flat surface," *International Journal of Heat and Mass Transfer*, vol. 82, pp. 316–324, 2015.
- [16] A. A. Avramenko, I. V. Shevchuk, S. Abdallah, D. G. Blinov, S. Harmand, and A. I. Tyrinov, "Symmetry analysis for film boiling of nanofluids on a vertical plate using a nonlinear approach," *Journal of Molecular Liquids*, vol. 223, pp. 156–164, 2016.
- [17] A. Malvandi, S. Heysiattalab, and D. D. Ganji, "Thermophoresis and Brownian motion effects on heat transfer enhancement at film boiling of nanofluids over a vertical cylinder," *Journal of Molecular Liquids*, vol. 216, pp. 503–509, 2016.
- [18] J. C. Maxwell, *A treatise on electricity and magnetism*. Clarendon press, 1873, vol. 1.
- [19] C. Qi, Y. L. Wan, C. Y. Li, D. T. Han, and Z. H. Rao, "Experimental and numerical research on the flow and heat transfer characteristics of TiO<sub>2</sub>-water nanofluids in a corrugated tube," *International Journal of Heat and Mass Transfer*, vol. 115, pp. 1072–1084, 2017.
- [20] S. U. S. Choi and J. A. Eastman, "Enhancing thermal conductivity of fluids with nanoparticles," Tech. Rep., 1995.
- [21] S. K. Das, N. Putra, P. Thiesen, and W. Roetzel, "Temperature dependence of thermal conductivity enhancement for nanofluids," *J. Heat Transfer*, vol. 125, no. 4, pp. 567–574, 2003.
- [22] G. Coccia, S. Tomassetti, and G. Di Nicola, "Thermal conductivity of nanofluids: A review of the existing correlations and a scaled semi-empirical equation," 11 2021.
- [23] M. Borzuei and Z. Baniamerian, "Role of nanoparticles on critical heat flux in convective boiling of nanofluids: Nanoparticle sedimentation and Brownian motion," *International Journal of Heat and Mass Transfer*, vol. 150, p. 119299, 2020.
- [24] G.-S. Wang, B. Song, and Z.-H. Liu, "Operation characteristics of cylindrical miniature grooved heat pipe using aqueous CuO nanofluids," *Experimental Thermal and Fluid Science*, vol. 34, no. 8, pp. 1415–1421, 2010.
- [25] M. B. B. Esfahani, S. M. Sajadi, N. H. Abu-Hamdeh, S. Bezzina, A. Abdollahi, A. Karimipour, F. Ghaemi, and D. Baleanu, "The effect of sedimentation phenomenon of the additives silver nano particles on water pool boiling heat transfer coefficient: A comprehensive experimental study," *Journal of Molecular Liquids*, vol. 345, p. 117891, 2022.
- [26] S. Rashidi, O. Mahian, and E. M. Languri, "Applications of nanofluids in condensing and evaporating systems: a review," *Journal of Thermal Analysis and Calorimetry*, vol. 131, pp. 2027–2039, 2018.
- [27] V. P. Carey, *Liquid-vapor phase-change phenomena: an introduction to the thermophysics of vaporization and condensation processes in heat transfer equipment*. CRC Press, 2020.

## References

- [28] X. Yin, C. Hu, M. Bai, and J. Lv, "Effects of depositional nanoparticle wettability on explosive boiling heat transfer: A molecular dynamics study," *International Communications in Heat and Mass Transfer*, vol. 109, p. 104390, 2019.
- [29] A. Kujawska, R. Mulka, S. Hamze, G. Żyła, B. Zajaczkowski, M. H. Buschmann, and P. Estellé, "The effect of boiling in a thermosyphon on surface tension and contact angle of silica and graphene oxide nanofluids," *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, vol. 627, p. 127082, 2021.
- [30] S. Vafaei, A. Purkayastha, A. Jain, G. Ramanath, and T. Borca-Tasciuc, "The effect of nanoparticles on the liquid–gas surface tension of Bi<sub>2</sub>Te<sub>3</sub> nanofluids," *Nanotechnology*, vol. 20, no. 18, p. 185702, 2009.
- [31] J. Ally, M. Kappl, H.-J. Butt, and A. Amirfazli, "Detachment force of particles from air liquid interfaces of films and bubbles," *Langmuir*, vol. 26, no. 23, pp. 18 135–18 143, 2010.
- [32] C. H. Chon, S. Paik, J. B. Tipton, and K. D. Kihm, "Effect of nanoparticle sizes and number densities on the evaporation and dryout characteristics for strongly pinned nanofluid droplets," *Langmuir*, vol. 23, no. 6, pp. 2953–2960, 2007.
- [33] S. Vafaei and D. Wen, "The effect of gold nanoparticles on the spreading of triple line," *Microfluidics and Nanofluidics*, vol. 8, pp. 843–848, 2010.
- [34] A. Chengara, A. D. Nikolov, D. T. Wasan, A. Trokhymchuk, and D. Henderson, "Spreading of nanofluids driven by the structural disjoining pressure gradient," *Journal of colloid and interface science*, vol. 280, no. 1, pp. 192–201, 2004.
- [35] D. T. Wasan and A. D. Nikolov, "Spreading of nanofluids on solids," *Nature*, vol. 423, no. 6936, pp. 156–159, 2003.
- [36] A. Nikolov, K. Kondiparty, and D. Wasan, "Nanoparticle self-structuring in a nanofluid film spreading on a solid surface," *Langmuir*, vol. 26, no. 11, pp. 7665–7670, 2010.
- [37] E. Abedini, A. Behzadmehr, S. M. Sarvari, and S. H. Mansouri, "Numerical investigation of subcooled flow boiling of a nanofluid," *International Journal of Thermal Sciences*, vol. 64, pp. 232–239, 2 2013.
- [38] S. N. Shoghl, M. Bahrami, and M. K. Moraveji, "Experimental investigation and CFD modeling of the dynamics of bubbles in nanofluid pool boiling," *International Communications in Heat and Mass Transfer*, vol. 58, pp. 12–24, 11 2014.
- [39] A. Abdollahi, H. A. Mohammed, S. M. Vanaki, A. Osia, and M. R. Golbahar Haghghi, "Fluid flow and heat transfer of nanofluids in microchannel heat sink with V-type inlet/outlet arrangement," *Alexandria Engineering Journal*, vol. 56, no. 1, pp. 161–170, 3 2017.
- [40] H. I. Mohammed, D. Giddings, and G. S. Walker, "CFD simulation of a concentrated salt nanofluid flow boiling in a rectangular tube," *International Journal of Heat and Mass Transfer*, vol. 125, pp. 218–228, 10 2018.
- [41] H. I. Mohammed, D. Giddings, G. S. Walker, P. Talebizadehsardari, and J. M. Mahdi, "Thermal behaviour of the flow boiling of a complex nanofluid in a rectangular channel: An experimental and numerical study," *International Communications in Heat and Mass Transfer*, vol. 117, p. 104773, 10 2020.

## References

- [42] A. Soleimani, A. Sattari, and P. Hanafizadeh, "Thermal analysis of a microchannel heat sink cooled by two-phase flow boiling of Al<sub>2</sub>O<sub>3</sub> HFE-7100 nanofluid," *Thermal Science and Engineering Progress*, vol. 20, 12 2020.
- [43] G. Li and X. Fang, "Numerical Simulation on the Boiling Flow Patterns of Al<sub>2</sub>O<sub>3</sub>-Water Nanofluid in Micro/Minichannel under Different Hypergravity Levels and Directions," *International Journal of Aerospace Engineering*, vol. 2021, 2021.
- [44] S. Zaboli, H. Alimoradi, and M. Shams, "Numerical investigation on improvement in pool boiling heat transfer characteristics using different nanofluid concentrations," *Journal of Thermal Analysis and Calorimetry*, pp. 1–18, 2022.
- [45] O. R. Alomar, S. R. Aslan, and F. G. Zaki, "Modelling and simulation of two-phase flow inside porous pipe evaporator using Cu-Water nano-fluid," *International Journal of Thermal Sciences*, vol. 175, p. 107462, 2022.
- [46] A. Mavi and T. Chinyoka, "Volume-of-Fluid Based Finite-Volume Computational Simulations of Three-Phase Nanoparticle-Liquid-Gas Boiling Problems in Vertical Rectangular Channels," *Energies*, vol. 15, no. 15, 8 2022.
- [47] A. Malvandi, S. A. Moshizi, E. G. Soltani, and D. D. Ganji, "Modified Buongiorno's model for fully developed mixed convection flow of nanofluids in a vertical annular pipe," *Computers & Fluids*, vol. 89, pp. 124–132, 2014.
- [48] F. Hedayati and G. Domairry, "Effects of nanoparticle migration and asymmetric heating on mixed convection of TiO<sub>2</sub>-H<sub>2</sub>O nanofluid inside a vertical microchannel," *Powder Technology*, vol. 272, pp. 250–259, 2015.
- [49] —, "Nanoparticle migration effects on fully developed forced convection of TiO<sub>2</sub>-water nanofluid in a parallel plate microchannel," *Particuology*, vol. 24, pp. 96–107, 2016.
- [50] E. E. Michaelides, "Brownian movement and thermophoresis of nanoparticles in liquids," *International Journal of Heat and Mass Transfer*, vol. 81, pp. 179–187, 2 2015.
- [51] A. A. Avramenko, D. G. Blinov, I. V. Shevchuk, and A. V. Kuznetsov, "Symmetry analysis and self-similar forms of fluid flow and heat-mass transfer in turbulent boundary layer flow of a nanofluid," *Physics of Fluids*, vol. 24, no. 9, 2012.
- [52] J. U. Brackbill, D. B. Kothe, and C. Zemach, "A continuum method for modeling surface tension," *Journal of computational physics*, vol. 100, no. 2, pp. 335–354, 1992.
- [53] C. S. Peskin, "Numerical analysis of blood flow in the heart," *Journal of computational physics*, vol. 25, no. 3, pp. 220–252, 1977.
- [54] C. Dopazo, "On conditioned averages for intermittent turbulent flows," *Journal of Fluid Mechanics*, vol. 81, no. 3, pp. 433–438, 1977.
- [55] H. G. Weller, "The development of a new flame area combustion model using conditional averaging," *Thermo-fluids section report TF*, vol. 9307, 1993.
- [56] H. Jasak, "A Consistent Derivation of the Sea-Ice Model Using Conditional Averaging," 2003.
- [57] J. Klostermann, K. Schaake, and R. Schwarze, "Numerical simulation of a single rising bubble by VOF with surface compression," *International Journal for Numerical Methods in Fluids*, vol. 71, no. 8, pp. 960–982, 2013.

## References

- [58] H. G. Weller, "A new approach to VOF-based interface capturing methods for incompressible, compressible and cavitating flow," in *Technical Report*. OpenCFD Limited, 2006.
- [59] E. Olsson and G. Kreiss, "A conservative level set method for two phase flow," *Journal of Computational Physics*, vol. 210, no. 1, pp. 225–246, 11 2005.
- [60] F. Habla, C. Fernandes, M. Maier, L. Densky, L. L. Ferrás, A. Rajkumar, O. S. Carneiro, O. Hinrichsen, and J. M. Nóbrega, "Development and validation of a model for the temperature distribution in the extrusion calibration stage," *Applied Thermal Engineering*, vol. 100, pp. 538–552, 5 2016.
- [61] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, "A tensorial approach to computational continuum mechanics using object-oriented techniques," *Computers in physics*, vol. 12, no. 6, pp. 620–631, 1998.
- [62] S. Hardt and F. Wondra, "Evaporation model for interfacial flows based on a continuum-field representation of the source terms," *Journal of Computational Physics*, vol. 227, no. 11, pp. 5871–5895, 2008.
- [63] A. A. Avramenko, I. V. Shevchuk, A. I. Tyrinov, and D. G. Blinov, "Heat transfer at film condensation of stationary vapor with nanoparticles near a vertical plate," *Applied Thermal Engineering*, vol. 73, no. 1, pp. 391–398, 2014.
- [64] B. v. Szyszkowski, "Experimentelle Studien über kapillare Eigenschaften der wässerigen Lösungen von Fettsäuren," *Zeitschrift für physikalische Chemie*, vol. 64, no. 1, pp. 385–414, 1908.
- [65] Y. Wang and J. Wu, "Numerical simulation on single bubble behavior during Al<sub>2</sub>O<sub>3</sub>/H<sub>2</sub>O nanofluids flow boiling using Moving Particle Simi-implicit method," *Progress in Nuclear Energy*, vol. 85, pp. 130–139, 6 2015.
- [66] A. Esmaeeli and G. Tryggvason, "Computations of film boiling. Part I: numerical method," *International journal of heat and mass transfer*, vol. 47, no. 25, pp. 5451–5461, 2004.
- [67] P. J. Berenson, "Film-boiling heat transfer from a horizontal surface," *J Heat Transf*, vol. 83, no. 3, p. 351–356, 1961.
- [68] A. A. Avramenko, D. G. Blinov, and I. V. Shevchuk, "Self-similar analysis of fluid flow and heat-mass transfer of nanofluids in boundary layer," *Physics of fluids*, vol. 23, no. 8, 2011.
- [69] S. Stephan and H. Hasse, "Interfacial properties of binary mixtures of simple fluids and their relation to the phase diagram," *Physical Chemistry Chemical Physics*, vol. 22, no. 22, pp. 12 544–12 564, 6 2020.
- [70] —, "Enrichment at vapour-liquid interfaces of mixtures: establishing a link between nanoscopic and macroscopic properties," pp. 319–349, 7 2020.
- [71] X. Yong, S. Qin, and T. J. Singler, "Nanoparticle-mediated evaporation at liquid-vapor interfaces," *Extreme Mechanics Letters*, vol. 7, pp. 90–103, 6 2016.

## References

# Chapter 5

## Conclusion

The chapter is dedicated to the conclusion of this Ph.D. project and includes a summary of the work as well as future works.

### 5.1 Summary

The research primarily focused on two objectives: firstly, studying different methods to enhance the performance of simulation techniques for thermal phase change phenomena; and secondly, improving the performance of boiling cooling technologies, with a particular emphasis on the use of nanofluids.

**Chapter 1** laid the foundation by introducing the importance of boiling techniques and reviewing a range of simulation methods, boiling instabilities, and enhancement strategies, including the adoption of nanofluids. This chapter emphasized the challenge in precisely identifying the gas-liquid interface in simulations, particularly highlighting the limitations of the VOF-MULES method, which is widely used in commercial and open-source CFD software. These limitations, such as smeared interfaces and the occurrence of non-physical velocities near the interface, are noted to arise in certain situations.

**Chapter 2** focused on the advancement of thermal phase change solvers within OpenFOAM, leading to the development of the ‘thermalPhaseChangeFlow’ solver. This solver incorporates the VOF-isoAdvector method, which prior research has shown to effectively minimize non-physical currents at the vapor-liquid interface. The ‘thermalPhaseChangeFlow’ solver demonstrated improvements over the twoPhaseFlow solver (another solver that can use isoAdvector in thermal phase change phenomena) in simulating varying contact angles. It exhibited equal (or slightly better) accuracy and less computational time compared to VOF-MULES in the studied thermal benchmark cases, minimized spurious currents, and achieved better precision in

curvature prediction, as demonstrated in the "Stationary 3D Spherical Bubble Scenario".

**Chapter 3** focused on implementing the CLSVOF method, which synergizes the Volume of Fluid (VOF) and level-set techniques to refine curvature calculation accuracy in thermal phase change scenarios. Recognized for its enhanced precision in interface curvature depiction, as noted in preceding studies, CLSVOF showed a marginally closer alignment with analytical models in thermal phase change benchmark scenarios, despite necessitating increased computational time compared to the standard VOF method. The detailed examination of the "Stationary 3D Spherical Bubble Scenario" further underscored CLSVOF's superior curvature prediction, although this came with the trade-off of higher spurious currents relative to the VOF methodology.

It is worth mentioning that, both chapters highlighted that in the 2D film boiling benchmark case, the VOF-isoAdvector and CLSVOF methods, when compared to VOF, present less accurate depictions of film boiling under conditions of coarse grid structures and closely aligned interfaces to the wall. It is important to note that the conclusions of this study are context-specific, based on the chosen conditions and schemes, and might differ in other scenarios<sup>1</sup>.

**Chapter 4** focused on the second objective by investigating the use of nanofluids to improve boiling cooling system performance. As mentioned in **Chapter 1** various instabilities in boiling cooling techniques can significantly impact efficiency. Among the strategies to mitigate these issues, the application of nanofluids is a promising approach for enhancing boiling heat transfer. Notably, in the film boiling regime—characterized by lower heat transfer rates due to the insulating vapor layer—nanofluids show potential for improved heat transfer. This chapter highlights that many CFD studies on nanofluid boiling omit a detailed governing equation for nanoparticle concentration, typically focusing only on the vapor phase and neglecting nanoparticle enrichment dynamics in both the liquid phase and at the vapor-liquid interface. The lack of a comprehensive model greatly limits the in-depth analysis and accurate simulation of nanoparticle enrichment at this critical interface. A novel approach was introduced, integrating a well-established governing equation for nanoparticle concentration with the Continuous-Species-Transfer (CST) method into the Computational Multi-Fluid Dynamics (CMFD) framework. This expanded the equation's application beyond single-phase scenarios to include both vapor and liquid phases and their interface. A key demonstration of this method was through a 2D axisymmetric film boiling analysis on a vertical cylinder. Among the key

---

<sup>1</sup>The solvers introduced in these studies are available at <https://github.com/AAU-OpenFOAM/LSThermalPhaseChangeFlow>

findings was the observation of increased nanoparticle concentration at the gas-liquid interface, a phenomenon commonly known as nanoparticle enrichment. To the best of the author's knowledge, this is the first CFD model of nanofluid boiling that successfully predicts this behavior. The study also noted substantial changes in nanofluids' thermophysical properties during film boiling, including increases in density, thermal conductivity, and kinematic viscosity, alongside a decrease in specific heat capacity and surface tension. Furthermore, the research established a link between nanoparticle concentration and heat transfer efficiency in nanofluid boiling, as evidenced by the variations in the space-averaged Nusselt number. These results underscore the improved heat transfer efficiency with higher nanoparticle concentrations in film boiling scenarios.

## 5.2 Future Works

While this Ph.D. thesis addresses specific challenges in boiling process modeling, there are additional aspects to consider, such as:

- This research examines the use of VOF-isoAdvector and CLSVOF instead of VOF-MULES to enhance the accuracy of thermal phase change solvers. Nevertheless, alternative interface description methods (e.g., phase field and PLIC) could be incorporated into a thermal phase change solver, or a combination of these methods could be implemented and optimized. This would provide access to various solvers with different interface description methods, each more suitable for specific flow conditions.
- The investigation into the use of VOF-isoAdvector and CLSVOF in thermal phase change solvers indicated only slight variations in results when compared to VOF-MULES. However, when the "Stationary 3D Spherical Bubble Scenario," a non-thermal phase change benchmark, was employed, more significant differences became apparent. While VOF-MULES is adequate in certain instances, its tendency to generate smeared interfaces can be problematic in other scenarios. In the specific thermal phase change benchmarks considered in this PhD research, VOF-MULES did not significantly affect the outcomes, making the advantages of VOF-isoAdvector and CLSVOF less discernible. This finding highlights the need for new thermal phase change benchmark cases that are specifically designed to underscore the improvements achievable with advanced methods like VOF-isoAdvector and CLSVOF.
- Further progress in nanofluid boiling simulation could include the development and implementation of a solution that demonstrates the variation of contact line due to the nanoparticle deposition on the base plate

and the influence of these variations on the hydraulic and thermal performance of nanofluid boiling.

- In this investigation, the ratio of  $\text{Al}_2\text{O}_3$  nanoparticles between the liquid water and vapor phases at their interface during boiling is hypothesized to be  $He = 8$ . The actual value of  $He$  for  $\text{Al}_2\text{O}_3$  nanoparticles, however, is not empirically established and requires experimental investigation for accurate determination. The precise calculation of  $He$  is not within the scope of this research, as it involves comprehensive experimental work to assess nanoparticle enrichment at the liquid-vapor interface and ascertain the appropriate  $He$  values.
- An important aspect yet to be fully explored in nanofluid boiling research is the experimental validation of modeling and simulation outcomes. The findings of computational modeling and simulations need to be corroborated with experimental data to ascertain their accuracy and reliability. Future work could hence focus on conducting systematic experimental studies, which would not only validate the computational models but also provide valuable insights into the physical phenomena under real-world conditions. This will significantly contribute to refining the existing models and crafting more accurate and reliable predictive tools for nanofluid boiling systems.

# Appendix A1

Listing A1.1: fvSolution File Employed For Stefan Problem

```
1 /*-----*- C++ -*-----*\
2 | ===== |
3 | \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4 | \\ / O p e r a t i o n | Version: v2006 |
5 | \\ / A n d | Website: www.openfoam.com |
6 | \\ M a n i p u l a t i o n |
7 /*-----*\
8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     object       fvSolution;
14 }
15 // ***** //
16
17 solvers
18 {
19     "(alpha.water.*|psi|H)"
20     {
21         advectionScheme isoAdvection;
22
23         interfaceMethod "INTERFACE_METHOD";
24         isoFaceTol      1e-10;
25         surfCellTol    1e-6;
26         nAlphaBounds   3;
27         snapTol        1e-12;
28         clip            true;
29
30         nAlphaCorr      2;
31         nAlphaSubCycles 1;
32         cAlpha          1;
33         reconstructionScheme isoAlpha; // plicRDF
34         advectionScheme isoAdvection;
35
36         MULESCorr       yes;
37     }
38 }
```

## Appendix A1.

```

38     maxUnboundedness 1e-5;
39     CoCoeff          2;
40     maxIter          5;
41     nLimiterIter     2;
42
43     solver            PBiCG;
44     preconditioner    DILU;
45     tolerance         1e-12;
46     relTol           0.01;
47 }
48
49 "(U|UFinal|T|TFinal)"
50 {
51     solver            PBiCG;
52     preconditioner    DILU;
53     tolerance         1e-06;
54     relTol           0;
55 };
56
57 rho
58 {
59     solver            diagonal;
60     tolerance         1e-7;
61     relTol           0.1;
62 }
63
64 rhoFinal
65 {
66     $rho;
67     tolerance         1e-7;
68     relTol           0;
69 }
70
71 alpha.waterFinal
72 {
73     solver            smoothSolver;
74     smoother          symGaussSeidel;
75     tolerance         1e-8;
76     relTol           0;
77     maxIter          10;
78 }
79
80 pcorr
81 {
82     $p;
83     relTol           0;
84 };
85
86 "(p_rgh|p)"
87 {
88     solver            PCG;
89     preconditioner
90     {
91         preconditioner GAMG;

```

```

92         tolerance      1e-08;
93         relTol         0;
94         nVcycles       2;
95         smoother       GaussSeidel;
96         nPreSweeps     0;
97         nPostSweeps   2;
98         nFinestSweeps 2;
99         cacheAgglomeration true;
100        nCellsInCoarsestLevel 10;
101        agglomerator   faceAreaPair;
102        mergeLevels    1;
103    }
104
105    tolerance      1e-08;
106    relTol         0.001;
107    maxIter        20;
108 };
109
110 "(p_rghFinal|pFinal)"
111 {
112     solver        PCG;
113     preconditioner
114     {
115         preconditioner GAMG;
116         tolerance      1e-08;
117         relTol         0;
118         nVcycles       2;
119         smoother       GaussSeidel;
120         nPreSweeps     2;
121         nPostSweeps   0;
122         nFinestSweeps 2;
123         cacheAgglomeration true;
124         nCellsInCoarsestLevel 10;
125         agglomerator   faceAreaPair;
126         mergeLevels    1;
127     }
128     tolerance      1e-08;
129     relTol         0;
130 };
131
132 mDotSmearFinal
133 {
134     solver        PCG;
135     tolerance      1e-6;
136     preconditioner DIC;
137     relTol         0.00;
138     smoother       DIC;
139 }
140
141
142 pcorrFinal
143 {
144     $p_rgh;
145     tolerance      1e-9;

```

## Appendix A1.

```

146     relTol        0;
147 }
148
149
150 "(h|T.*|k|epsilon|R)"
151 {
152     solver        smoothSolver;
153     smoother      symGaussSeidel;
154     tolerance     1e-7;
155     relTol        0.;
156     minIter       15;
157     maxIter       50;
158 }
159
160 "(U|h|T.*|k|epsilon|R)Final"
161 {
162     $U;
163     tolerance     1e-7;
164     relTol        0;
165     maxIter       50;
166 }
167
168 Phi
169 {
170     $p_rgh;
171     relTol        0;
172 };
173 }
174
175 PIMPLE
176 {
177     momentumPredictor    true;
178     nOuterCorrectors     1;
179     nCorrectors          3;
180     nNonOrthogonalCorrectors  0;
181
182     nAlphaCorr           1;
183     nAlphaSubCycles     4;
184     cAlpha               1;
185
186     pRefCell             0;
187     pRefValue            0.0;
188
189     smoothItr            3.0;
190     kSmoothItr           1.0;
191 }
192
193 relaxationFactors
194 {
195     equations
196     {
197         ".*"            1;
198     }
199 }

```



## Appendix A1.

Listing A1.2: controlDict File Employed For Stefan Problem

```

1 /*-----*- C++ -*-----*\
2 | ===== |
3 | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \ / O p e r a t i o n | Version: v2006 |
5 | \ \ / A n d | Website: www.openfoam.com |
6 | \ \ / M a n i p u l a t i o n | |
7 /*-----*\
8 FoamFile
9 {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "system";
14     object        controlDict;
15 }
16 // ***** //
17
18 application      LSThermalPhaseChangeFlow;
19
20 startFrom        latestTime;
21
22 startTime        0;
23
24 stopAt           endTime;
25
26 endTime          50;
27
28 deltaT           1e-7;
29
30 writeControl     adjustableRunTime;
31
32 writeInterval    10;
33
34 purgeWrite       0;
35
36 writeFormat      ascii;
37
38 writePrecision   6;
39
40 writeCompression off;
41
42 timeFormat       general;
43
44 timePrecision    6;
45
46 runtimeModifiable yes;
47
48 adjustTimeStep   yes;
49
50 maxCo            0.5;
51
52 maxAlphaCo       0.5;

```

```

53
54 maxDeltaT      2e-4;
55
56
57 functions
58 {
59     interfaceParameters
60     {
61         type interfaceParameters;
62         libs ("libinterfaceParametersFunctionObject.so");
63         phase alpha.water;
64         writeControl writeTime;
65         writeToFile true;
66         enabled true;
67     }
68
69     stefanInterface
70     {
71         // Load the library containing the 'coded' functionObject
72         libs ("libutilityFunctionObjects.so");
73         type coded;
74         writeControl adjustableRunTime;
75         writeInterval 1;
76
77         // Name of on-the-fly generated functionObject
78         name stefanInterface;
79
80         // List of include search directories
81         codeOptions
82         #{
83             -I$(LIB_SRC)/sampling/lnInclude \
84             -I$(LIB_SRC)/surfMesh/lnInclude
85         #};
86
87         // List of libraries to link with
88         codeLibs
89         #{
90             -lsampling \
91             -lsurfMesh
92         #};
93
94         // List of include files
95         codeInclude
96         #{
97             #include "sampledIsoSurfaceCell.H"
98         #};
99
100        // Code
101        codeWrite
102        #{
103            // Compute isosurface alpha=0.5
104            dictionary isoSurfDict;
105            isoSurfDict.add("type", "isoSurfaceCell");
106            isoSurfDict.add("isoField", "alpha.water");

```

## Appendix A1.

```

107     isoSurfDict.add("isoValue",0.5);
108     isoSurfDict.add("interpolate",false);
109     sampledIsoSurfaceCell isoInterface("isoInterface", mesh(),
isoSurfDict);
110     isoInterface.update();
111
112     // Compute position of the isosurface
113     const pointField& pf = isoInterface.points();
114     scalar xinter = 0.002;
115     forAll(pf, i)
116     {
117         xinter = min(pf[i][0],xinter);
118     }
119
120     // Print results
121     Info << "Interface position (x) = " << xinter << "\n" << endl;
122     #};
123 }
124
125 #includeFunc components(U)
126 #includeFunc "writeCellCentres"
127
128 line1
129 {
130     type                sets;
131     libs                (sampling);
132     executeControl      onEnd;
133     writeControl        adjustableRunTime;
134     writeInterval       1;
135     interpolationScheme  cellPoint;
136     setFormat           raw;
137     sets
138     (
139         line
140         {
141             type        uniform;
142             axis        x;
143             start       (0 1e-5 1e-5);
144             end         (2e-3 1e-5 1e-5);
145             nPoints     400;
146         }
147     );
148     fields (T);
149 }
150 }
151
152 // ***** //

```

Listing A1.3: fvSolution File Employed For Horizontal Film Condensation

```

1  /*-----*- C++ -*-----*/
2  |=====|
3  | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \ / O p e r a t i o n | Version: v2006 |
5  | \ \ / A n d | Website: www.openfoam.com |
6  | \ \ / M a n i p u l a t i o n | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     object        fvSolution;
14 }
15 // ***** //
16
17 solvers
18 {
19     "(alpha.water.*|psi|H)"
20     {
21         interfaceMethod "INTERFACE_METHOD";
22         isoFaceTol      1e-10;
23         surfCellTol     1e-6;
24         nAlphaBounds    3;
25         snapTol         1e-12;
26         clip            true;
27
28         nAlphaCorr      2;
29         nAlphaSubCycles 1;
30         cAlpha          1;
31         reconstructionScheme isoAlpha; // plicRDF
32         advectionScheme isoAdvection;
33
34         MULESCorr       yes;
35
36         maxUnboundedness 1e-5;
37         CoCoeff          2;
38         maxIter          5;
39         nLimiterIter     2;
40
41         solver           PBiCG;
42         preconditioner   DILU;
43         tolerance        1e-12;
44         relTol           0.01;
45     }
46
47     "(U|UFinal|T|TFinal)"
48     {
49         solver           PBiCG;
50         preconditioner   DILU;
51         tolerance        1e-06;
52         relTol           0;

```

## Appendix A1.

```

53 };
54
55 rho
56 {
57     solver          diagonal;
58     tolerance       1e-7;
59     relTol          0.1;
60 }
61
62 rhoFinal
63 {
64     $rho;
65     tolerance       1e-7;
66     relTol          0;
67 }
68
69 alpha.waterFinal
70 {
71     solver          smoothSolver;
72     smoother        symGaussSeidel;
73     tolerance       1e-8;
74     relTol          0;
75     maxIter         10;
76 }
77
78 pcorr
79 {
80     $p;
81     relTol          0;
82 };
83
84 "(p_rgh|p)"
85 {
86     solver          PCG;
87     preconditioner
88     {
89         preconditioner  GAMG;
90         tolerance       1e-08;
91         relTol          0;
92         nVcycles        2;
93         smoother        GaussSeidel;
94         nPreSweeps      0;
95         nPostSweeps     2;
96         nFinestSweeps   2;
97         cacheAgglomeration true;
98         nCellsInCoarsestLevel 10;
99         agglomerator    faceAreaPair;
100        mergeLevels     1;
101    }
102
103    tolerance       1e-08;
104    relTol          0.001;
105    maxIter         20;
106 };

```

```

107
108 "(p_rghFinal|pFinal)"
109 {
110     solver          PCG;
111     preconditioner
112     {
113         preconditioner  GAMG;
114         tolerance       1e-08;
115         relTol          0;
116         nVcycles        2;
117         smoother        GaussSeidel;
118         nPreSweeps      2;
119         nPostSweeps     0;
120         nFinestSweeps   2;
121         cacheAgglomeration true;
122         nCellsInCoarsestLevel 10;
123         agglomerator    faceAreaPair;
124         mergeLevels     1;
125     }
126     tolerance        1e-08;
127     relTol           0;
128 };
129
130 mDotSmearFinal
131 {
132     solver          PCG;
133     tolerance       1e-6;
134     preconditioner  DIC;
135     relTol          0.00;
136     smoother        DIC;
137 }
138
139
140 pcorrFinal
141 {
142     $p_rgh;
143     tolerance       1e-9;
144     relTol          0;
145 }
146
147
148 "(h|T.*|k|epsilon|R)"
149 {
150     solver          smoothSolver;
151     smoother        symGaussSeidel;
152     tolerance       1e-7;
153     relTol          0.;
154     minIter         15;
155     maxIter         50;
156 }
157
158 "(U|h|T.*|k|epsilon|R)Final"
159 {
160     $U;

```

## Appendix A1.

```
161     tolerance      1e-7;
162     relTol        0;
163     maxIter       50;
164 }
165
166 Phi
167 {
168     $p_rgh;
169     relTol        0;
170 };
171 }
172
173 PIMPLE
174 {
175     momentumPredictor      true;
176     nOuterCorrectors       1;
177     nCorrectors             3;
178     nNonOrthogonalCorrectors 0;
179
180     nAlphaCorr              1;
181     nAlphaSubCycles         4;
182     cAlpha                  1;
183
184     pRefCell                0;
185     pRefValue               0.0;
186
187     smoothItr               3.0;
188     kSmoothItr              1.0;
189 }
190
191 relaxationFactors
192 {
193     equations
194     {
195         ".*"              1;
196     }
197 }
198
199 // ***** //
```

Listing A1.4: controlDict File Employed For Horizontal Film Condensation

```

1  /*-----*- C++ -*-----*\
2  | ===== |
3  | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \ / O p e r a t i o n | Version: v2006 |
5  | \ \ / A n d | Website: www.openfoam.com |
6  | \ \ / M a n i p u l a t i o n | |
7  /*-----*\
8  FoamFile
9  {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     location     "system";
14     object       controlDict;
15 }
16 // ***** //
17
18 application     LSThermalPhaseChangeFlow;
19
20 startFrom       latestTime;
21
22 startTime       0;
23
24 stopAt          endTime;
25
26 endTime         4;
27
28 deltaT          1E-08;
29
30 writeControl    adjustableRunTime;
31
32 writeInterval   0.25;
33
34 purgeWrite      0;
35
36 writeFormat     ascii;
37
38 writePrecision  6;
39
40 writeCompression off;
41
42 timeFormat      general;
43
44 timePrecision   6;
45
46 runtimeModifiable yes;
47
48 adjustTimeStep  yes;
49
50 maxCo           0.01;
51
52 maxAlphaCo      0.01;

```

## Appendix A1.

```
53
54 maxDeltaT      1E-05;
55
56 functions
57 {
58
59     interfaceParameters
60     {
61         type interfaceParameters;
62         libs ("libinterfaceParametersFunctionObject.so");
63         phase alpha.water;
64         writeControl writeTime;
65         writeToFile true;
66         enabled true;
67     }
68
69     interfaceHeight1
70     {
71         // Mandatory entries
72         type          interfaceHeight;
73         libs          (fieldFunctionObjects);
74         locations     ((0 0 0));
75
76
77         // Optional entries
78         alpha        alpha.water;
79         direction     (1 0 0);
80         interpolationScheme cellPoint;
81
82         // Optional (inherited) entries
83         writePrecision 16;
84
85         enabled      true;
86         log          true;
87         timeStart    0;
88         timeEnd      5;
89         writeControl writeTime;
90     }
91 }
92 // * * * * * //
```

Listing A1.5: fvSolution File Employed For Film Condensation on a Vertical Plate

```

1  /*-----*- C++ -*-----*/
2  |=====|
3  | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \ / O p e r a t i o n | Version: v2006 |
5  | \ \ / A n d | Website: www.openfoam.com |
6  | \ \ / M a n i p u l a t i o n | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     object        fvSolution;
14 }
15 // ***** //
16
17 solvers
18 {
19     "(alpha.water.*|psi|H)"
20     {
21         interfaceMethod "INTERFACE_METHOD";
22         isoFaceTol      1e-10;
23         surfCellTol     1e-6;
24         nAlphaBounds    3;
25         snapTol         1e-12;
26         clip            true;
27
28         nAlphaCorr      2;
29         nAlphaSubCycles 1;
30         cAlpha          1;
31         reconstructionScheme isoAlpha; // plicRDF
32         advectionScheme isoAdvection;
33
34
35         MULESCorr       yes;
36
37         maxUnboundedness 1e-5;
38         CoCoeff         2;
39         maxIter         5;
40         nLimiterIter    2;
41
42         solver          PBiCG;
43         preconditioner  DILU;
44         tolerance       1e-12;
45         relTol          0.01;
46     }
47
48     "(U|UFinal|T|TFinal)"
49     {
50         solver          PBiCG;
51         preconditioner  DILU;
52         tolerance       1e-06;

```

## Appendix A1.

```

53     relTol          0;
54 };
55
56 rho
57 {
58     solver          diagonal;
59     tolerance       1e-7;
60     relTol          0.1;
61 }
62
63 rhoFinal
64 {
65     $rho;
66     tolerance       1e-7;
67     relTol          0;
68 }
69
70 alpha.waterFinal
71 {
72     solver          smoothSolver;
73     smoother        symGaussSeidel;
74     tolerance       1e-8;
75     relTol          0;
76     maxIter         10;
77 }
78
79 pcorr
80 {
81     $p;
82     relTol          0;
83 };
84
85 "(p_rgh|p)"
86 {
87     solver          PCG;
88     preconditioner
89     {
90         preconditioner GAMG;
91         tolerance     1e-08;
92         relTol        0;
93         nVcycles      2;
94         smoother      GaussSeidel;
95         nPreSweeps    0;
96         nPostSweeps   2;
97         nFinestSweeps 2;
98         cacheAgglomeration true;
99         nCellsInCoarsestLevel 10;
100        agglomerator   faceAreaPair;
101        mergeLevels    1;
102    }
103
104    tolerance        1e-08;
105    relTol            0.001;
106    maxIter           20;

```

```

107 };
108
109 "(p_rghFinal|pFinal)"
110 {
111     solver          PCG;
112     preconditioner
113     {
114         preconditioner  GAMG;
115         tolerance       1e-08;
116         relTol          0;
117         nVcycles        2;
118         smoother        GaussSeidel;
119         nPreSweeps      2;
120         nPostSweeps     0;
121         nFinestSweeps   2;
122         cacheAgglomeration true;
123         nCellsInCoarsestLevel 10;
124         agglomerator    faceAreaPair;
125         mergeLevels     1;
126     }
127     tolerance        1e-08;
128     relTol           0;
129 };
130
131 mDotSmearFinal
132 {
133     solver          PCG;
134     tolerance       1e-6;
135     preconditioner  DIC;
136     relTol          0.00;
137     smoother        DIC;
138 }
139
140
141 pcorrFinal
142 {
143     $p_rgh;
144     tolerance       1e-9;
145     relTol          0;
146 }
147
148
149 "(h|T.*|k|epsilon|R)"
150 {
151     solver          smoothSolver;
152     smoother        symGaussSeidel;
153     tolerance       1e-7;
154     relTol          0.;
155     minIter         15;
156     maxIter         50;
157 }
158
159 "(U|h|T.*|k|epsilon|R)Final"
160 {

```

## Appendix A1.

```
161     $U;
162     tolerance      1e-7;
163     relTol         0;
164     maxIter        50;
165 }
166
167 Phi
168 {
169     $p_rgh;
170     relTol         0;
171 };
172 }
173
174 PIMPLE
175 {
176     momentumPredictor      true;
177     nOuterCorrectors        1;
178     nCorrectors              3;
179     nNonOrthogonalCorrectors 0;
180
181     nAlphaCorr               1;
182     nAlphaSubCycles          4;
183     cAlpha                   1;
184
185     pRefCell                  0;
186     pRefValue                 0.0;
187
188     smoothItr                 3.0;
189     kSmoothItr                1.0;
190 }
191
192 relaxationFactors
193 {
194     equations
195     {
196         ".*"                 0.5;
197     }
198 }
199
200 // ***** //
```

Listing A1.6: controlDict File Employed For Film Condensation on a Vertical Plate

```

1 /*-----*- C++ -*-----*\
2 | ===== |
3 | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \ / O p e r a t i o n | Version: v2006 |
5 | \ \ / A n d | Website: www.openfoam.com |
6 | \ \ / M a n i p u l a t i o n | |
7 /*-----*\
8 FoamFile
9 {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "system";
14     object        controlDict;
15 }
16 // ***** //
17
18 application      LSThermalPhaseChangeFlow;
19
20 startFrom        latestTime;
21
22 startTime        0;
23
24 stopAt           endTime;
25
26 endTime          1;
27
28 deltaT           1E-08;
29
30 writeControl     adjustableRunTime;
31
32 writeInterval    0.1;
33
34 purgeWrite       0;
35
36 writeFormat      ascii;
37
38 writePrecision   6;
39
40 writeCompression off;
41
42 timeFormat       general;
43
44 timePrecision    6;
45
46 runtimeModifiable yes;
47
48 adjustTimeStep   yes;
49
50 maxCo            0.1;
51
52 maxAlphaCo       0.1;

```

## Appendix A1.

```
53
54 maxDeltaT      1E-03;
55
56 functions
57 {
58     interfaceParameters
59     {
60         type interfaceParameters;
61         libs ("libinterfaceParametersFunctionObject.so");
62         phase alpha.water;
63         writeControl writeTime;
64         writeToFile true;
65         enabled true;
66     }
67
68     isosurfaces
69     {
70         type surfaces;
71         libs ("libsampling.so");
72         writeControl writeTime;
73         surfaceFormat raw;
74         fields ( alpha.water );
75
76         surfaces
77         (
78             interpolatedIso
79             {
80                 type isoSurface;
81                 isoField alpha.water;
82                 isoValue 0.5;
83                 interpolate true;
84                 regularise false;
85             }
86         );
87     }
88 }
89
90 // ***** //
```

Listing A1.7: fvSolution File Employed For 2D Film Boiling

```

1  /*-----*- C++ -*/
2  |=====|
3  | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \ / O p e r a t i o n | Version: v2006 |
5  | \ \ / A n d | Website: www.openfoam.com |
6  | \ \ / M a n i p u l a t i o n | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     object        fvSolution;
14 }
15 // ***** //
16
17 solvers
18 {
19     "(alpha.water.*|psi|H)"
20     {
21         interfaceMethod "INTERFACE_METHOD";
22         isoFaceTol      1e-10;
23         surfCellTol     1e-6;
24         nAlphaBounds    3;
25         snapTol         1e-12;
26         clip            true;
27
28         nAlphaCorr      2;
29         nAlphaSubCycles 1;
30         cAlpha          1;
31         reconstructionScheme isoAlpha; // plicRDF
32         advectionScheme isoAdvection;
33
34
35         MULESCorr       yes;
36
37         maxUnboundedness 1e-5;
38         CoCoeff         2;
39         maxIter         5;
40         nLimiterIter    2;
41
42         solver          PBiCG;
43         preconditioner  DILU;
44         tolerance       1e-12;
45         relTol          0.01;
46     }
47
48     "(U|UFinal|T|TFinal)"
49     {
50         solver          PBiCG;
51         preconditioner  DILU;
52         tolerance       1e-06;

```

## Appendix A1.

```

53     relTol          0;
54 };
55
56 rho
57 {
58     solver          diagonal;
59     tolerance       1e-7;
60     relTol         0.1;
61 }
62
63 rhoFinal
64 {
65     $rho;
66     tolerance       1e-7;
67     relTol         0;
68 }
69
70 alpha.waterFinal
71 {
72     solver          smoothSolver;
73     smoother        symGaussSeidel;
74     tolerance       1e-8;
75     relTol         0;
76     maxIter        10;
77 }
78
79 pcorr
80 {
81     $p;
82     relTol         0;
83 };
84
85 "(p_rgh|p)"
86 {
87     solver          PCG;
88     preconditioner
89     {
90         preconditioner GAMG;
91         tolerance     1e-08;
92         relTol       0;
93         nVcycles     2;
94         smoother     GaussSeidel;
95         nPreSweeps   0;
96         nPostSweeps  2;
97         nFinestSweeps 2;
98         cacheAgglomeration true;
99         nCellsInCoarsestLevel 10;
100        agglomerator  faceAreaPair;
101        mergeLevels   1;
102    }
103
104    tolerance       1e-08;
105    relTol         0.001;
106    maxIter        20;

```

```

107 };
108
109 "(p_rghFinal|pFinal)"
110 {
111     solver          PCG;
112     preconditioner
113     {
114         preconditioner  GAMG;
115         tolerance       1e-08;
116         relTol         0;
117         nVcycles       2;
118         smoother       GaussSeidel;
119         nPreSweeps     2;
120         nPostSweeps    0;
121         nFinestSweeps  2;
122         cacheAgglomeration true;
123         nCellsInCoarsestLevel 10;
124         agglomerator   faceAreaPair;
125         mergeLevels    1;
126     }
127     tolerance       1e-08;
128     relTol         0;
129 };
130
131 mDotSmearFinal
132 {
133     solver          PCG;
134     tolerance       1e-6;
135     preconditioner  DIC;
136     relTol         0.00;
137     smoother       DIC;
138 }
139
140
141 pcorrFinal
142 {
143     $p_rgh;
144     tolerance       1e-9;
145     relTol         0;
146 }
147
148
149 "(h|T.*|k|epsilon|R)"
150 {
151     solver          smoothSolver;
152     smoother       symGaussSeidel;
153     tolerance       1e-7;
154     relTol         0.;
155     minIter        15;
156     maxIter        50;
157 }
158
159 "(U|h|T.*|k|epsilon|R)Final"
160 {

```

## Appendix A1.

```

161     $U;
162     tolerance      1e-7;
163     relTol        0;
164     maxIter       50;
165 }
166
167 Phi
168 {
169     $p_rgh;
170     relTol        0;
171 };
172 }
173
174 PIMPLE
175 {
176     momentumPredictor      true;
177     nOuterCorrectors        1;
178     nCorrectors              3;
179     nNonOrthogonalCorrectors 0;
180
181     nAlphaCorr               1;
182     nAlphaSubCycles          4;
183     cAlpha                   1;
184
185     pRefCell                 0;
186     pRefValue                 0.0;
187
188     smoothItr                 3.0;
189     kSmoothItr                1.0;
190 }
191
192 relaxationFactors
193 {
194     equations
195     {
196         ".*"                  1;
197     }
198 }
199
200 // ***** //

```

Listing A1.8: controlDict File Employed For 2D Film Boiling

```

1 /*-----*- C++ -*-----*\
2 | ===== |
3 | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \ / O p e r a t i o n | Version: v2006 |
5 | \ \ / A n d | Website: www.openfoam.com |
6 | \ \ / M a n i p u l a t i o n | |
7 /*-----*\
8 FoamFile
9 {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "system";
14     object        controlDict;
15 }
16 // ***** //
17
18 application      LSThermalPhaseChangeFlow;
19
20 startFrom        latestTime;
21
22 startTime        0;
23
24 stopAt           endTime;
25
26 endTime          1;
27
28 deltaT           1e-18;
29
30 writeControl     adjustableRunTime;
31
32 writeInterval    0.01;
33
34 purgeWrite       0;
35
36 writeFormat      ascii;
37
38 writePrecision   6;
39
40 writeCompression off;
41
42 timeFormat       general;
43
44 timePrecision    8;
45
46 runtimeModifiable yes;
47
48 adjustTimeStep   yes;
49
50 maxCo            0.1;
51
52 maxAlphaCo       0.1;

```

## Appendix A1.

```

53
54 maxDeltaT      0.1;
55
56 functions
57 {
58     interfaceParameters
59     {
60         type interfaceParameters;
61         libs ("libinterfaceParametersFunctionObject.so");
62         phase alpha.water;
63         writeControl writeTime;
64         writeToFile true;
65         enabled true;
66     }
67
68
69     // *****
70     // *** coded functionObject for calculating Berenson number *****
71     // *****
72
73     NusseltCorrelation
74     {
75         // Load the library containing the 'coded' functionObject
76         libs ("libutilityFunctionObjects.so");
77         type coded;
78         writeControl adjustableRunTime;
79         writeInterval 0.001;
80         // Name of on-the-fly generated functionObject
81         name analyticalSolution;
82         // Code
83         codeWrite
84         #{
85         scalar sigma_ = 0.07e-03;
86         scalar rhog_ = 242.7;
87             scalar rhol_ = 402.4;
88             scalar g_ = 9.81;
89             scalar Hlg_ = 276.4e03;
90             scalar kg_ = 0.538;
91             scalar mug_ = 32.38e-6;
92             scalar DT_ = 5;
93             scalar Cpg_ = 3.52e5;
94             scalar lambda_ = sqrt( sigma_/((rhol_-rhog_)*g_) );
95             const volScalarField& T=mesh().lookupObject<volScalarField>("T");
96             // Compute Berenson's correlation number
97             scalar Nub_ = 0.425*pow(((rhog_*(rhol_-rhog_)*g_*Hlg_)/(kg_*mug_*abs
(DT_))),0.25)*pow(lambda_,0.75);
98             Info << "Berenson correlation number = " << Nub_ << endl;
99             // Compute Klimenko's correlation number
100             scalar Beta_ = Cpg_ * DT_ / Hlg_;
101             scalar Pr_ = Cpg_ * mug_ / kg_;
102             scalar Gr_ = pow(rhog_,2)*g_*pow(lambda_,3)/pow(mug_,2)*(rhol_/rhog_

```

```

-1);
103     scalar Nuk_ = 0.19*pow(Gr_,0.33333333)*pow(Pr_,0.333333)*0.89*pow(
Beta_,-0.333333);
104     Info << "Klimenko correlation number = " << Nuk_ << endl;
105     label down = mesh().boundary().findPatchID("down");
106
107     volScalarField Nusselt
108     (
109         IOobject
110         (
111             "Nusselt",
112             mesh().time().timeName(),
113             mesh(),
114             IOobject::NO_READ,
115             IOobject::AUTO_WRITE
116         ),
117         mesh(),
118         dimensionedScalar("Nusselt", dimless, 0.0)
119     );
120     Nusselt.boundaryFieldRef()[down] = lambda_/DT_*T.boundaryField()[
down].snGrad();
121     scalar area = gSum(mesh().magSf().boundaryField()[down]);
122     scalar avgNusselt = gSum(Nusselt.boundaryField()[down] * mesh().
magSf().boundaryField()[down])/area;
123     Info << "Space-averaged Nusselt lambda = " << avgNusselt << "\n" <<
endl;
124     #};
125 }
126 }
127
128 // ***** //

```

## Appendix A1.

**Listing A1.9:** fvSolution File Employed For Stationary 3D Spherical Bubble

```

1  /*-----*- C++ -*-----*/
2  |=====|
3  | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \ / O p e r a t i o n | Version: v2006 |
5  | \ \ / A n d | Website: www.openfoam.com |
6  | \ \ / M a n i p u l a t i o n | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     object        fvSolution;
14 }
15 // ***** //
16
17 solvers
18 {
19     "(alpha.water.*|psi|H)"
20     {
21         interfaceMethod "isoAdvector";
22         isoFaceTol      1e-10;
23         surfCellTol     1e-6;
24         nAlphaBounds    3;
25         snapTol         1e-12;
26         clip             true;
27
28         nAlphaCorr      2;
29         nAlphaSubCycles 1;
30         cAlpha          1;
31         reconstructionScheme plicRDF;
32         advectionScheme isoAdvection;
33
34
35         MULESCorr       yes;
36
37         maxUnboundedness 1e-5;
38         CoCoeff         2;
39         maxIter         5;
40         nLimiterIter    2;
41
42         solver          PBiCG;
43         preconditioner  DILU;
44         tolerance       1e-12;
45         relTol          0.01;
46     }
47
48     "(U|UFinal|T|TFinal)"
49     {
50         solver          PBiCG;
51         preconditioner  DILU;
52         tolerance       1e-06;

```

```

53     relTol          0;
54 };
55
56 rho
57 {
58     solver          diagonal;
59     tolerance       1e-7;
60     relTol          0.1;
61 }
62
63 rhoFinal
64 {
65     $rho;
66     tolerance       1e-7;
67     relTol          0;
68 }
69
70 alpha.waterFinal
71 {
72     solver          smoothSolver;
73     smoother        symGaussSeidel;
74     tolerance       1e-8;
75     relTol          0;
76     maxIter         10;
77 }
78
79 pcorr
80 {
81     $p;
82     relTol          0;
83 };
84
85 "(p_rgh|p)"
86 {
87     solver          PCG;
88     preconditioner
89     {
90         preconditioner  GAMG;
91         tolerance       1e-08;
92         relTol          0;
93         nVcycles        2;
94         smoother        GaussSeidel;
95         nPreSweeps      0;
96         nPostSweeps     2;
97         nFinestSweeps   2;
98         cacheAgglomeration true;
99         nCellsInCoarsestLevel 10;
100        agglomerator     faceAreaPair;
101        mergeLevels      1;
102    }
103
104    tolerance         1e-08;
105    relTol             0.001;
106    maxIter            20;

```

## Appendix A1.

```

107 };
108
109 "(p_rghFinal|pFinal)"
110 {
111     solver          PCG;
112     preconditioner
113     {
114         preconditioner  GAMG;
115         tolerance       1e-08;
116         relTol         0;
117         nVcycles       2;
118         smoother       GaussSeidel;
119         nPreSweeps     2;
120         nPostSweeps    0;
121         nFinestSweeps  2;
122         cacheAgglomeration true;
123         nCellsInCoarsestLevel 10;
124         agglomerator   faceAreaPair;
125         mergeLevels    1;
126     }
127     tolerance       1e-08;
128     relTol         0;
129 };
130
131 mDotSmearFinal
132 {
133     solver          PCG;
134     tolerance       1e-6;
135     preconditioner  DIC;
136     relTol         0.00;
137     smoother       DIC;
138 }
139
140
141 pcorrFinal
142 {
143     $p_rgh;
144     tolerance       1e-9;
145     relTol         0;
146 }
147
148
149 "(h|T.*|k|epsilon|R)"
150 {
151     solver          smoothSolver;
152     smoother       symGaussSeidel;
153     tolerance       1e-7;
154     relTol         0.;
155     minIter        15;
156     maxIter        50;
157 }
158
159 "(U|h|T.*|k|epsilon|R)Final"
160 {

```

```

161     $U;
162     tolerance      1e-7;
163     relTol        0;
164     maxIter       50;
165 }
166
167 Phi
168 {
169     $p_rgh;
170     relTol        0;
171 };
172 }
173
174 PIMPLE
175 {
176     momentumPredictor      true;
177     nOuterCorrectors       1;
178     nCorrectors            3;
179     nNonOrthogonalCorrectors 0;
180
181     nAlphaCorr             1;
182     nAlphaSubCycles       4;
183     cAlpha                 1;
184
185     pRefCell               0;
186     pRefValue              0.0;
187
188     smoothItr              3.0;
189     kSmoothItr             1.0;
190 }
191
192 relaxationFactors
193 {
194     equations
195     {
196         ".*"              1;
197     }
198 }
199
200 // ***** //

```

## Appendix A1.

Listing A1.10: controlDict File Employed For Stationary 3D Spherical Bubble

```

1  /*-----*- C++ -*-----*\
2  | ===== |
3  | \ \      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \      / O p e r a t i o n | Version: v2006 |
5  | \ \      / A n d | Website: www.openfoam.com |
6  | \ \      / M a n i p u l a t i o n | |
7  /*-----*- C++ -*-----*\
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "system";
14     object        controlDict;
15 }
16 // ***** //
17
18 application      LSThermalPhaseChangeFlow;
19
20 startFrom        latestTime;
21
22 startTime        0;
23
24 stopAt           endTime;
25
26 endTime          1;
27
28 deltaT           1e-18;
29
30 writeControl      adjustableRunTime;
31
32 writeInterval     0.01;
33
34 purgeWrite        0;
35
36 writeFormat       ascii;
37
38 writePrecision    6;
39
40 writeCompression  off;
41
42 timeFormat         general;
43
44 timePrecision      8;
45
46 runtimeModifiable yes;
47
48 adjustTimeStep    yes;
49
50 maxCo              0.1;
51
52 maxAlphaCo         0.1;

```

```

53
54 maxDeltaT      0.1;
55
56 functions
57 {
58     interfaceParameters
59     {
60         type interfaceParameters;
61         libs ("libinterfaceParametersFunctionObject.so");
62         phase alpha.water;
63         writeControl writeTime;
64         writeToFile true;
65         enabled true;
66     }
67
68
69     // *****
70     // *** coded functionObject for calculating Berenson number *****
71     // *****
72
73     NusseltCorrelation
74     {
75         // Load the library containing the 'coded' functionObject
76         libs ("libutilityFunctionObjects.so");
77         type coded;
78         writeControl adjustableRunTime;
79         writeInterval 0.001;
80         // Name of on-the-fly generated functionObject
81         name analyticalSolution;
82         // Code
83         codeWrite
84         #{
85         scalar sigma_ = 0.07e-03;
86         scalar rhog_ = 242.7;
87             scalar rhol_ = 402.4;
88             scalar g_ = 9.81;
89             scalar Hlg_ = 276.4e03;
90             scalar kg_ = 0.538;
91             scalar mug_ = 32.38e-6;
92             scalar DT_ = 5;
93             scalar Cpg_ = 3.52e5;
94             scalar lambda_ = sqrt( sigma_/((rhol_-rhog_)*g_) );
95             const volScalarField& T=mesh().lookupObject<volScalarField>("T");
96             // Compute Berenson's correlation number
97             scalar Nub_ = 0.425*pow(((rhog_*(rhol_-rhog_)*g_*Hlg_)/(kg_*mug_*abs
(DT_))),0.25)*pow(lambda_,0.75);
98             Info << "Berenson correlation number = " << Nub_ << endl;
99             // Compute Klimenko's correlation number
100             scalar Beta_ = Cpg_ * DT_ / Hlg_;
101             scalar Pr_ = Cpg_ * mug_ / kg_;
102             scalar Gr_ = pow(rhog_,2)*g_*pow(lambda_,3)/pow(mug_,2)*(rhol_/rhog_

```

## Appendix A1.

```

-1);
103     scalar Nuk_ = 0.19*pow(Gr_,0.33333333)*pow(Pr_,0.333333)*0.89*pow(
Beta_,-0.333333);
104     Info << "Klimenko correlation number = " << Nuk_ << endl;
105     label down = mesh().boundary().findPatchID("down");
106
107     volScalarField Nusselt
108     (
109         IOobject
110         (
111             "Nusselt",
112             mesh().time().timeName(),
113             mesh(),
114             IOobject::NO_READ,
115             IOobject::AUTO_WRITE
116         ),
117         mesh(),
118         dimensionedScalar("Nusselt", dimless, 0.0)
119     );
120     Nusselt.boundaryFieldRef()[down] = lambda_/DT_*T.boundaryField()[
down].snGrad();
121     scalar area = gSum(mesh().magSf().boundaryField()[down]);
122     scalar avgNusselt = gSum(Nusselt.boundaryField()[down] * mesh().
magSf().boundaryField()[down])/area;
123     Info << "Space-averaged Nusselt lambda = " << avgNusselt << "\n" <<
endl;
124     #};
125 }
126 }
127
128 // ***** //

```



