UNIVERSITÄT STUTTGART

Numerical Modelling of Evaporation and Condensation Phenomena

Numerische Modellierung von Verdampfungs- und Kondensationsphänomenen

IRS-13-S57
MASTER THESIS

by cand. aer. Jibran HAIDER

July 2013

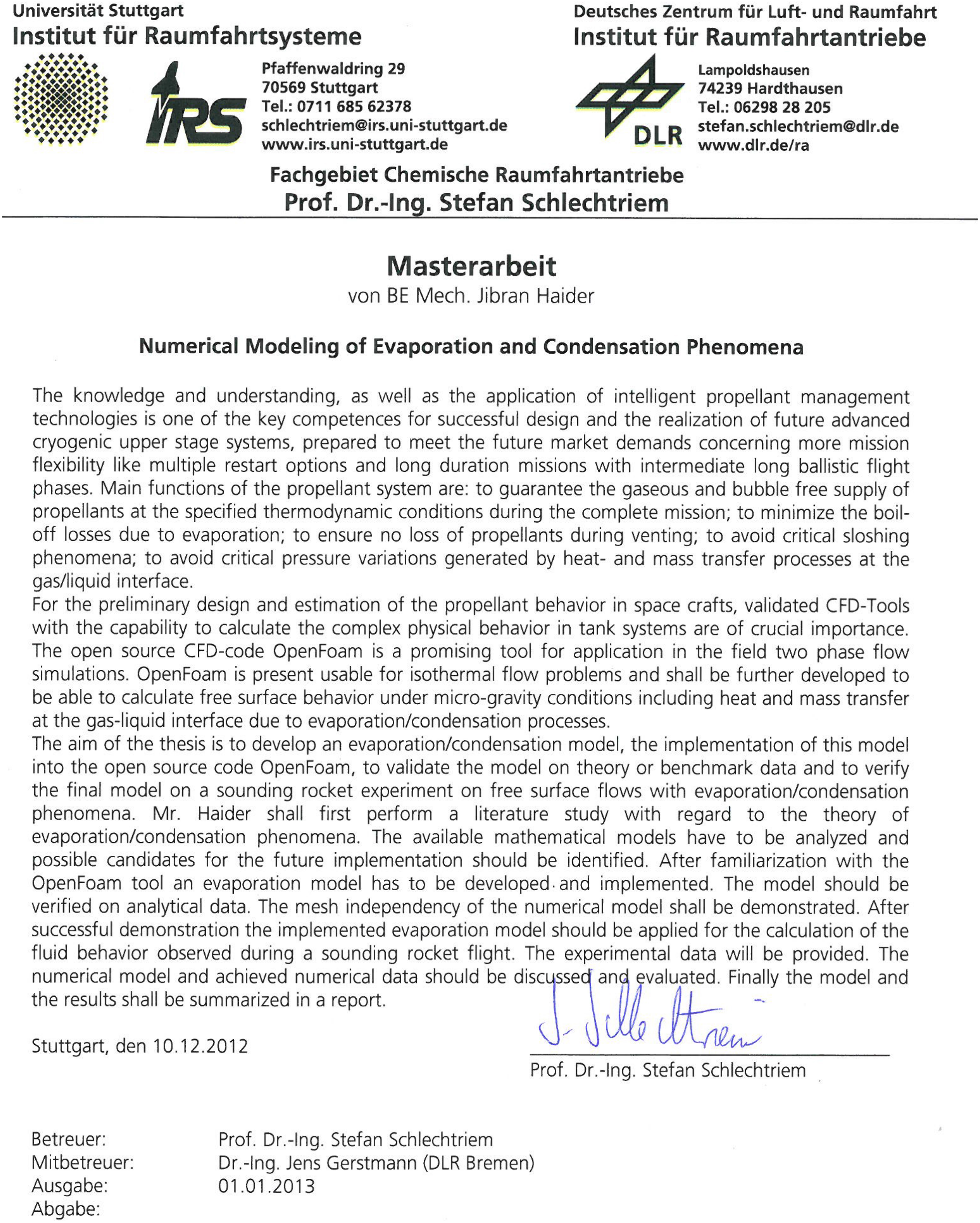
Supervisors:

Dr.-Ing. Jens GERSTMANN
Prof. Dr.-Ing. Stefan SCHLECHTRIEM

Deutschen Zentrum für Luft und Raumfahrt,
Institut für Raumfahrtantriebe (RA)

Universität Stuttgart,
Institut für Raumfahrtsysteme (IRS)

Task Description



Universität Stuttgart
Institut für Raumfahrtsysteme

Pfaffenwaldring 29
70569 Stuttgart
Tel.: 0711 685 62378
schlechtriem@irs.uni-stuttgart.de
www.irs.uni-stuttgart.de

Deutsches Zentrum für Luft- und Raumfahrt
Institut für Raumfahrtantriebe

Lampoldshausen
74239 Hardthausen
Tel.: 06298 28 205
stefan.schlechtriem@dlr.de
www.dlr.de/ra

Fachgebiet Chemische Raumfahrtantriebe
Prof. Dr.-Ing. Stefan Schlechtriem

Masterarbeit

von BE Mech. Jibran Haider

Numerical Modeling of Evaporation and Condensation Phenomena

The knowledge and understanding, as well as the application of intelligent propellant management technologies is one of the key competences for successful design and the realization of future advanced cryogenic upper stage systems, prepared to meet the future market demands concerning more mission flexibility like multiple restart options and long duration missions with intermediate long ballistic flight phases. Main functions of the propellant system are: to guarantee the gaseous and bubble free supply of propellants at the specified thermodynamic conditions during the complete mission; to minimize the boil-off losses due to evaporation; to ensure no loss of propellants during venting; to avoid critical sloshing phenomena; to avoid critical pressure variations generated by heat- and mass transfer processes at the gas/liquid interface.

For the preliminary design and estimation of the propellant behavior in space crafts, validated CFD-Tools with the capability to calculate the complex physical behavior in tank systems are of crucial importance. The open source CFD-code OpenFoam is a promising tool for application in the field two phase flow simulations. OpenFoam is present usable for isothermal flow problems and shall be further developed to be able to calculate free surface behavior under micro-gravity conditions including heat and mass transfer at the gas-liquid interface due to evaporation/condensation processes.

The aim of the thesis is to develop an evaporation/condensation model, the implementation of this model into the open source code OpenFoam, to validate the model on theory or benchmark data and to verify the final model on a sounding rocket experiment on free surface flows with evaporation/condensation phenomena. Mr. Haider shall first perform a literature study with regard to the theory of evaporation/condensation phenomena. The available mathematical models have to be analyzed and possible candidates for the future implementation should be identified. After familiarization with the OpenFoam tool an evaporation model has to be developed and implemented. The model should be verified on analytical data. The mesh independency of the numerical model shall be demonstrated. After successful demonstration the implemented evaporation model should be applied for the calculation of the fluid behavior observed during a sounding rocket flight. The experimental data will be provided. The numerical model and achieved numerical data should be discussed and evaluated. Finally the model and the results shall be summarized in a report.

Stuttgart, den 10.12.2012

Prof. Dr.-Ing. Stefan Schlechtriem

Betreuer:	Prof. Dr.-Ing. Stefan Schlechtriem
Mitbetreuer:	Dr.-Ing. Jens Gerstmann (DLR Bremen)
Ausgabe:	01.01.2013
Abgabe:	

Declaration

I, Jibran Haider, declare that this Masters Thesis titled, 'Numerical Modelling of Evaporation and Condensation Phenomena' and the work presented in this report are my own. I confirm that:

- this work was done wholly in candidature for a research degree at University of Stuttgart.
- the research was carried out at the German Aerospace Center (DLR), in Transport and Propulsion Systems (TAS) department at DLR's site in Bremen, Germany.
- wherever previously published work of others has been included in this report, it has always been attributed.
- acknowledgements of all the sources which have helped me in the compilation of this report have been done.

Signed:

____

Dated:

____

Abstract

Investigation of interfacial phase change phenomena has been a subject of keen interest due to the complexities involved in the evaporation and condensation processes. Therefore, a numerical model based on the Volume of Fluid (VOF) method has been developed in OpenFOAM software package. This model is capable of simulating evaporation and condensation phenomena at a liquid-vapor interface subjected to non-isothemal boundary conditions. This is part of the study to simulate the phase change phenomena in HydroFluoroEther (HFE-7000) as observed in the Sounding Rocket Compere Experiment 2 (SOURCE 2). SOURCE 2 focuses on the investigation of behaviour of propellants for future cryogenic space missions. The phase change model is first applied to a benchmark evaporation model. The numerical results are in well agreement with theoretical results. Finally, the model was verified on the sounding rocket experiment (SOURCE 2) and a satisfactory comparison of results was obtained.

Keywords: Interfacial Phase Change, Multiphase Flow, Evaporation, Condensation, Numerical Modelling, OpenFOAM, Computational Fluid Dynamics

Kurzfassung

Die Untersuchung des Phänomens der Phasenänderung ist wegen der Komplexität von Verdampfungs- und Kondensationsprozessen vom großen Interesse. Dafür wurde ein nummerisches Volume of Fluid (VOF) Modell in der OpenFOAM Software Umgebung entwickelt. Mit diesem Modell ist es möglich, die Phänomene der Verdampfung und Kondensation an der Phasengrenze mit nicht-isothermen Randbedingungen zu simulieren. Ein Teil dieser Arbeit fließt in die Simulation der Phasennderung des HydroFluoroEther (HFE-7000) ein, welche im Sounding Rocket Compere Experiment 2 (SOURCE 2) beobachtet wurde. SOURCE 2 konzentriert sich auf die Untersuchung des Verhaltens von Treibstoffen für zukünftige kryogene Raumfahrtmissionen. Das Phasenänderungsmodell wurde vorerst an einem Verdampfungsmodell bewertet. Die nummerischen Ergebnisse stimmen sehr gut mit den theoretischen Ergebnissen überein. Am Ende der Arbeit wurde das Modell mit SOURCE 2 verifiziert und ein zufriedenstellender Vergleich der Ergebnisse erreicht.

Schlagwärter: Phasenänderung, Mehrphasenströmung, Verdampfung, Kondensation, nummerisches Modellierung, Computational Fluid Dynamics (CFD)

Acknowledgements

I would like to extend my sincerest gratitude to my supervisor at DLR Bremen, Dr.-Ing. Jens Gerstmann for his valuable time and inputs at crucial moments. I would also like to thank Mr. Malte Stief for his help in introduction to Linux environment and OpenFOAM. Last but not the least, I would thank Prof. Dr.-Ing Stefan Schlechtriem for supervising my thesis at the university and Dipl. Ing. Bernd Wagner for providing the required assistance.

Contents

Task Description i

Declaration ii

Abstract iii

Acknowledgements iv

List of Figures viii

List of Tables x

Abbreviations xi

Nomenclature xii

1 Introduction 1

1.1 Background and Motivation 1

1.2 Multiphase Flows 2

1.2.1 Interfacial Phase Change Phenomena 2

1.2.2 Interface Capturing Methods 3

1.3 OpenFOAM 6

1.4 General Remarks 7

2 Governing Equations 8

2.1 Phase Transport Equations 9

2.2 Conservation of Mass 13

2.3 Conservation of Linear Momentum 13

2.4 Conservation of Energy 15

2.5 Two Phase Properties 18

3 Interfacial Phase Change Phenomena 19

3.1 Mathematical Model 19

3.2 Implementation in OpenFOAM 23

3.2.1 Existing Solvers 23

3.2.2 Development of a New Solver 24

4 Benchmark Evaporation Case 25

4.1 Evaporation Model 25

4.2 Fluid Properties 26

4.3 Grid Independence Study (GIS) 28

4.4 Results and Discussion 30

5 Sounding Rocket Compere Experiment 2 35

5.1 Experimental Details 35

5.1.1 First Phase 36

5.1.2 Second Phase 37

6 Numerical Study 38

6.1 Geometry 38

6.2 Meshing 40

6.3 Initial Conditions 41

6.3.1 Tank r1 41

6.4 Boundary Conditions 41

6.4.1 Walls 41

6.4.2 Inlets 42

6.4.3 Axis of Symmetry 42

6.4.4 Summary 43

6.5 Working Fluid 45

6.6 Pressure Velocity Coupling 45

6.7 Settings 46

6.8 Discetization Procedures 46

7 Results and Discussion 47

7.1 Phase Distribution 48

7.2 Temperature Distribution 50

7.3 Liquid-Vapor Ratio 59

8 Summary and Outlook 60

A 1D Heat Transfer through an Interface 62

B Tutorials 64

B.1 Geometry Creation in Gmsh 64

B.2 Modifying OpenFOAM Solvers 66

B.3 Modifying OpenFOAM Libraries 68

C HFE 7000 Properties 70

D Codes 73

D.1 Source 2 Geometry 73

D.2 interEvapCondPhaseChangeFoam Solver 83

E Derivation of Clausius-Clapeyron Relation 87

F Expansion of Terms 88

List of Figures

1.1 Volume of Fluid approach . . . 5

2.1 Control volume at liquid-vapor interface . . . 8

3.1 Evaporation and condensation process . . . 19

4.1 Benchmark evaporation model . . . 25

4.2 Rates of evaporation for GIS of BEC . . . 29

4.3 Interface height for BEC at $\Delta H_v = 30KJ/kg$ . . . 30

4.4 Interface height for BEC at $\Delta H_v = 40KJ/kg$ . . . 31

4.5 Interface height for BEC at $\Delta H_v = 50KJ/kg$ . . . 32

4.6 Comparison of numerical and theoretical evaporation rates for benchmark case . . . 33

4.7 Actual C_e for BEC . . . 34

5.1 SOURCE 2 test cell at the start of experiment . . . 35

5.2 SOURCE 2 test cell at t = 13 seconds . . . 36

5.3 SOURCE 2 test cell at t = 20 seconds . . . 36

5.4 SOURCE 2 test cell at t = 42 seconds . . . 37

6.1 Flowchart summarising numerical methodology . . . 38

6.2 SOURCE 2 geometry with dimensions . . . 39

6.3 Axi-symmetric SOURCE 2 geometry as modelled in Gmsh . . . 39

6.4 SOURCE 2 test cell meshing in Gmsh . . . 40

6.5 SOURCE 2 test cell meshing in the vicinity of GPPS . . . 40

7.1 Comparison of experimental and numerical interface position during the 1st phase . . . 48

7.2 Interface position obtained numerically during 2nd phase for case 1 . . . 49

7.3 Temperature contours at 0 seconds for case 1 50

7.4 Temperature contours at 0, 10 and 15 seconds for case 1 51

7.5 Temperature contours at 20, 25 and 30 seconds for case 1 52

7.6 Temperature contours at 35, 40 and 42 seconds for case 1 53

7.7 Temperature contours near the interface at 20 and 25 seconds for case 1 . 54

7.8 Temperature contours near the interface at 30 and 35 seconds for case 1 . 54

7.9 Temperature contours near the interface at 40 and 42 seconds for case 1 . 55

7.10 Temperature variation at TC-12 56

7.11 Temperature variation at TC-18 57

7.12 Temperature variation at TC-30 58

7.13 Liquid-vapor ratio for various accommodation coefficients 59

A.1 1D Heat Transfer 62

List of Tables

4.1 General properties of benchmark fluid 26

4.2 Phase properties of benchmark fluid 27

4.3 Grid sizes for GIS of evaporation benchmark model 28

4.4 Rates of evaporation for GIS ($C_e = 0.06$) 28

4.5 Actual C_e for BEC 33

6.1 Types of BCs implemented in OpenFOAM for SOURCE 2 44

6.2 Phase properties of HFE 7000 at RTP 45

7.1 Accommodation coefficients for numerical simulations 47

7.2 Thermocouple locations in SOURCE 2 experiment 55

Abbreviations

BC	**B**oundary **C**onditions
BEC	**B**enchmark **E**vaporation **C**ase
CFD	**C**omputational **F**luid **D**ynamics
DLR	**D**eutschen Zentrum für **L**uft und **R**aumfahrt
FVM	**F**inite **V**olume **M**ethod
GIS	**G**rid **I**ndependence **S**tudy
GPPS	**G**as **P**ort **P**hase **S**eparator
HFE	**H**ydro**F**luoro**E**ther
KTG	**K**inetic **T**heory of **G**ases
LHS	**L**eft **H**and **Side**
LSM	**L**evel **S**et **M**ethod
MCLSM	**M**ass **C**onserving **L**evel **S**et **M**ethod
OpenFOAM	**Open** Source **F**ield **O**peration **A**nd **M**anipulation
RHS	**R**ight **H**and **S**ide
RTP	**R**oom **T**emperature and **P**ressure
SOURCE	**SOU**nding **R**ocket **C**ompere **E**xperiment
VOF	**V**olume **O**f **F**luid

Nomenclature

Vectors and Tensors

$\boldsymbol{F}$	Diffusive flux vector	
$\boldsymbol{H}$	Height vector	$[m]$
$\boldsymbol{\mathsf{I}}$	Identity tensor	
$\boldsymbol{n}$	Normal vector	$[m]$
$\boldsymbol{U}$	Fluid velocity vector	$[m\,s^{-1}]$
$\boldsymbol{W}$	Control volume velocity vector	$[m\,s^{-1}]$

Latin Characters

c	Specific heat	$[J\,mol^{-1}\,K^{-1}]$
C	Accommodation coefficient	
Co	Courant number	
e	Specific energy	$[J\,kg^{-1}]$
h	Specific enthalpy	$[J\,kg^{-1}]$
$\mathit{\Delta} H$	Latent heat	$[J\,kg^{-1}]$
j	Flux of molecules	
k	Thermal conductivity	$[W\,m^{-1}\,K^{-1}]$
m	Mass of a molecule	$[kg]$
$\dot{m}$	Rate of mass transfer	$[kg\,m^{3}\,s^{-1}]$
M	Molar mass	$[kg\,mol^{-1}]$
p	Thermodynamic pressure	$[N\ m^{-2}]$
p_rgh	Pressure excluding hydrodynamic pressure	$[N\,m^{-2}]$
q	Heat flux	$[W\,m^{-2}]$
S	Surface	$[m^{2}]$
t	Time	$[s]$

T	Temperature	$[K]$
v	Specific volume	$[m^3\,kg^{-1}]$
V	Volume	$[m^3]$
x	Distance	$[m]$

Greek Characters

α	Phase fraction	
ρ	Density	$[kg\,m^{-3}]$
$\dot{\rho}$	Rate of change of density	$[kg\,m^{-3}]$
μ	Dynamic viscosity	$[Pa\,s]$
ν	Kinematic viscosity	$[m^2\,s^{-1}]$
$\boldsymbol{\sigma}$	Stress tensor	$[Pa]$
κ	Curvature	
$\mathit{\Gamma}$	Distribution function	
$\mathit{\Psi}$	Source term	
$\boldsymbol{\mathit{\Phi}}$	Flux density vector	
λ	Bulk viscosity	

Subscripts

For any arbitrary quantity 'Q' the subscript represents Q:

Q_l	in liquid phase
Q_v	in vapour phase
Q_c	during condensation
Q_e	during evaporation
Q_r	relatively
Q_i	at interface
Q_{cr}	at critical point
Q_{sat}	at saturation conditions

Q_w	at wall
Q_s	at surface
Q_v	during vaporization
Q_a	in phase a
Q_t	in total
Q_m	mechanically
Q_{r1}	in tank r1 of SOURCE 2 test cell
Q_{tib}	at top inlet boundary of SOURCE 2 test cell
Q_{r1w}	at the wall of tank r1 of SOURCE 2 test cell

Superscripts

For any arbitrary quantity 'Q' the superscript represents Q:

Q^T	after taking a transpose (Tensor)

Operators

$\nabla \cdot ()$	Divergence
$\nabla()$	Gradient
$\frac{D()}{Dt}$	Material Derivative
$\frac{d()}{dt}$	Local Derivative

Chapter 1

Introduction

1.1 Background and Motivation

Study of the phase change phenomena is of prime importance in many industrial applications, including aerospace industry, mainly due to the complexities involved in the phase change process. The applications in space where microgravity conditions prevail, the study becomes even more vital and has revealed some fascinating experimental results.

The knowledge and application of intelligent propellant management technologies is one of the key competences for successful design and the realization of future advanced cryogenic upper stage systems. This is necessary to meet the future market demands concerning more mission flexibility like multiple restart options and long duration missions with intermediate long ballistic flight phases. Some of the characteristics desirable in future propellant system includes:

- a continuous, gaseous free supply of the propellant during the complete mission.
- minimize the propellant loss due to evaporation.
- avoid propellant loss during venting process.
- avoid critical sloshing phenomena.
- avoid critical pressure variations arising due to heat and mass transfer.

1.2 Multiphase Flows

In multiphase flows, the phases are separated by interfaces which are subjected to deformation during the flow. The interface is a very thin layer, which separates the two phases and has properties which are distinct from the bulk phases. Determination of the variation of properties across the interface poses the greatest challenge to model multiphase flows.

In numerical analysis, modelling of the interface can be either done by a sharp or a smeared interface. In case of a sharp interface, a jump in properties would occur as we move from one phase to the other. However, for a smeared interface, which typically covers a few grid cells, a continuous transition in properties is experienced across the interface. Both methods have their share of advantages and disadvantages.

Generally, in CFD it is not practical to achieve an interface thickness of the orders of 10^{-9}, thereby resulting in the smearing of the interface over a few grid cells in the vicinity of the expected interface location.

1.2.1 Interfacial Phase Change Phenomena

Investigation of the interfacial evaporation and condensation has long been a prime subject of empirical study purely because of the complexities involved in the phase change phenomena. However, after a century long research to fully understand it, no fundamental method has been widely accepted. The problem is commonly associated to the definition of properties at the liquid-vapor interface.

The three methods used for studying it are through Continuum Mechanics, Kinetic Theory of Gases (KTG) and Statistical Rate Theory [1]. The more recent Statistical Rate Theory method has shown promise, however the KTG has been widely used as the method to study interfacial phase change phenomena.

The earliest experimental investigation can be dated back to the late 19th century when Hertz investigated the phase change rate for Mercury. He came to the conclusion that for every specie, there exists a maximum rate of phase change which depends on the temperature of the surface and the properties of the specie. Since then several experiments were carried out which consistently showed that the observed rate was significantly less than the maximum phase change rate. This led to the introduction of the term 'accommodation coefficient' by Knudsen, to account for the deviation between the maximum and observed phase change rates. The result was the famous Hertz-Knudsen equation to determine the net rate of phase change. Further modifications, discussed in Chapter 4, were done resulting in the Hertz-Knudsen-Schrage equation.

The study carried out by Zhang and Wang [2] focuses on the development of a modified thermodynamic interfacial phase change expression of a liquid under the effects of capillarity. A critical analysis has been carried out on the previous phase change models by mentioning their limitations. The confusions over the use of a correct phase change model has been attributed to the negligence between saturated and practical vapor pressure in thermodynamic non-equilibrium. The study shows that the concave liquid surface will increase vaporization and a convex liquid surface will promote condensation.

1.2.2 Interface Capturing Methods

Several numerical methods are available in literature to track interfaces. The research paper by Wörner [3] provides a detailed insight on the numerical methods and models for interface capturing in multiphase flows.

Interface tracking methods can broadly be divided into three categories, namely [4] [5]

1. Surface Tracking Methods
2. Volume Tracking Methods
3. Moving Mesh Methods

Of all these methods the Level Set Method (LSM), a sub-category of Surface Tracking Methods, and the Volume of Fluid Method (VOF), a sub-category of Volume Tracking Methods, have been the more popular ones. Both have their share of advantages and disadvantages, however the VOF method has been widely used in recent past.

Surface Tracking Methods

Initial developments in Surface Tracking Methods can be dated back to 1969, when Daly [5] presented a method to track the interface by a set of marker particles. The position of these markers during the multiphase flow was tracked which was used to determine the interface. Later in 1973, Hirt and Nichols [6] extended the idea based on height functions. However more lately in 1988, Osher and Sethian [7] introduced the more popular LSM which utilizes a distance function of the numerical cell to the interface. It works on the Eulerian approach.

Volume Tracking Methods

Early developments in Marker and Cell Method (MAC) can be dated back to 1965, when Harlow and Welch [8] introduced this concept based on marker particles spread over the entire fluid domain. In this method, the interface is marked by weightless particles which are transported convectively by the velocity. Cells with no marker particles are considered empty while those with marker particles are considered to be occupied by fluid. In between these two types of cells lies the interface. Later, the idea was extended to track the interface based on phase fractions in the VOF method which is discussed in detail below.

VOF Method:

The Volume of Fluid Method (VOF) is a numerical technique based on Eulerian approach to track and locate free surface in a two phase flow. It was first reported by Hirt and Nichols in 1975 and later more completely by them in 1981 [9]. It was shown that this method was more efficient and flexible than the other methods.

The principle behind the working of VOF method is the definition of a phase fraction field (α), which has a value between '1' and '0'. The value of unity corresponds to any point occupied by Fluid 'A' and zero otherwise. The average value of α in a cell would indicate the fractional volume of the cell occupied by the liquid. In the VOF method, the sum of the fractional volumes for all 'n' phases is equal to unity.

$$\sum_{a=1}^{n} \left(\frac{V_a}{V_t} \right) = 1 \tag{1.1}$$

Therefore, a cell with an average value of $\alpha = 1$ implies that it is completely occupied by Fluid A. Similarly, a cell with an average value of $\alpha = 0$ suggests that Fluid A is not present there. If any cell has an average value between unity and zero, it means that an interface exists in that region. This is illustrated in the Figure 1.1.

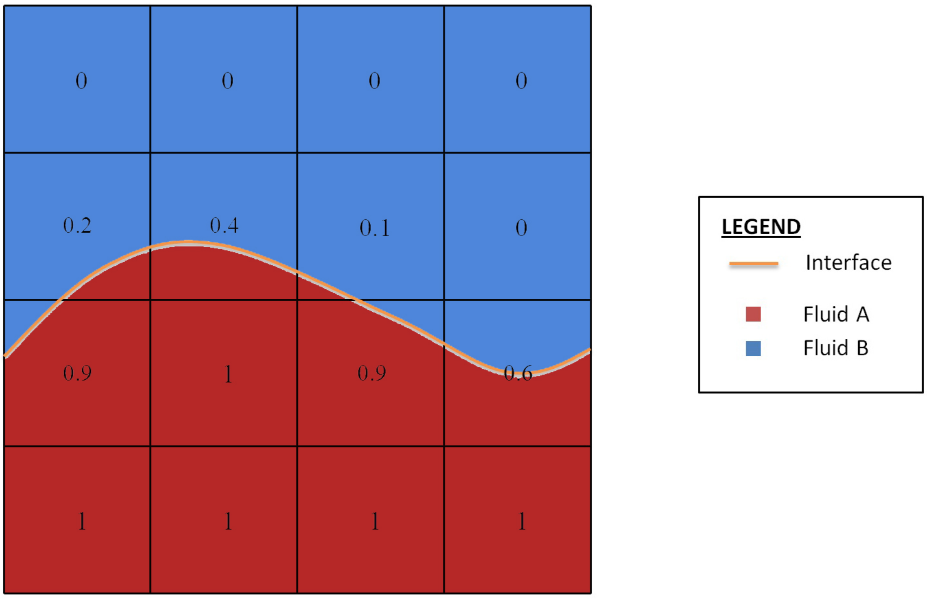


Figure 1.1: Volume of Fluid approach

Problems are encountered near the interface where there is a jump in the fluid properties. Therefore correct relations in the interface for fluid properties are vital for accurate modelling. It is one of the most widely used methods and is the one chosen for carrying out the two-phase flow simulations in the present study.

1.3 OpenFOAM

OpenFOAM (Open Source Field Operation and Manipulation) is an open source software package written in C++ for the solution of Continuum Mechanics problems based on Finite Volume Method (FVM), in particularly CFD. Continuum Mechanics is a branch of mechanics concerned with the stresses in solids, liquids and gases along with their flow and deformation. It's initial development can be dated back to late 1980s at Imperial College, London. Mr. Hrvoje Jasak and Mr. Henry Weller are the two most important personalities associated with the development of OpenFOAM.

The PhD thesis of H. Jasak [10], provides a detailed insight into the working of the Finite Volume Method associated with OpenFOAM. Quite a few changes have taken place since then but the basic foundations remain the same.

Later the study carried out by Weller et al. [11] aimed at the development of an object oriented computational continuum mechanics code. The approach presented in this paper has led to the creation of a C++ class library which makes it possible to implement complicated mathematical and physical models. The top level syntax of the code closely resembles the standard vector and tensor notation. The object oriented programming methodology allows the production of a code that is easier to write, validate and maintain than conventional procedural codes such as in Fortran. Object oriented techniques such as abstraction, inheritance, polymorphism and operator overloading have been effectively utilized in the creation of the code which is able to solve incompressible, compressible, multiphase and free surface flows together with turbulence modelling.

A study to evaluate the interface capturing algorithm by Raees et.al [12] discusses the advantages of a Mass Conserving Level Set Method (MCLSM) which combines the efficiency of a Level Set Method and the mass conserving properties of the Volume of Fluid method. The interface capturing algorithm implemented in OpenFOAM has been evaluated for incompressible and immiscible two phase simulations. It is concluded that the non conventional compressive scheme approach which is implemented in OpenFOAM, greatly reduces computation time as geometric reconstruction is not implemented. It is also found that highly accurate results will be predicted by OpenFOAM for the interface position if very fine meshes are used.

1.4 General Remarks

Some remarks regarding the thesis are mentioned below:

- All terms exclusive to OpenFOAM are written in *italics* throughout the document.
- All properties are expressed in SI units specified in the 'Nomenclature' unless otherwise stated.
- Vectors and Tensors are represented by **bold** characters.
- All simulations concerning this thesis were run on OpenFOAM version 2.1.1.

Chapter 2

Governing Equations

In order to define the governing equations for multiphase flow of this kind we need to consider a control volume at the liquid-vapor interface. This is shown in Figure 2.1. Conservation of an arbitrary quantity 'b' in the control volume can be represented as:

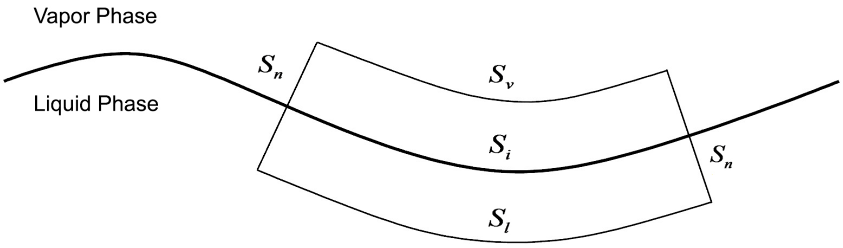


FIGURE 2.1: Control volume at liquid-vapor interface

$$\frac{d}{dt}\int\limits_{V(t)} b\ dV = \int\limits_{S(t)} -\Phi \cdot \boldsymbol{n}_s\ dS + \int\limits_{V(t)} \Psi\ dV \tag{2.1}$$

The term on the LHS of the equation is the time rate of change of quantity 'b' inside the control volume 'V'. The terms on the RHS represent the net inflow of quantity 'b' through all of the surfaces 'S' of the control volume and the net production rate inside the control volume. By considering that the control volume is fixed in time we can represent the flux density of 'b' as:

$$\Phi = b\boldsymbol{U} + \boldsymbol{F} \tag{2.2}$$

where '$\boldsymbol{U}$' is the fluid velocity and '$\boldsymbol{F}$' are the diffusive fluxes. By inserting in Eq (2.1) we obtain:

$$\frac{d}{dt}\int\limits_{V(t)} b\; dV = \int\limits_{S(t)} -(b\boldsymbol{U} + \boldsymbol{F}) \cdot \boldsymbol{n}_s\; dS + \int\limits_{V(t)} \Psi\; dV \tag{2.3}$$

Applying the Divergence Theorem results in:

$$\frac{d}{dt}\int\limits_{V(t)} b\; dV = \int\limits_{V(t)} \left[-\nabla \cdot (b\boldsymbol{U}) - \nabla \cdot \boldsymbol{F} + \Psi\right]\; dV \tag{2.4}$$

Now we can convert it to a partial differential equation.

$$\frac{\partial b}{\partial t} + \nabla \cdot (b\boldsymbol{U} + \boldsymbol{F}) = \Psi \tag{2.5}$$

$$\frac{\partial b}{\partial t} + \boldsymbol{U} \cdot \nabla b + b\nabla \cdot \boldsymbol{U} + \nabla \cdot \boldsymbol{F} = \Psi \tag{2.6}$$

Eq (2.6) represents a partial differential form for the conservation of any quantity 'b' inside a control volume. It will be used now to define the governing equations involved to simulate a non isothermal, multiphase flow needed for the development of the desired phase change solver. The governing equations are stated in this chapter and a suitable form to be implemented in OpenFOAM is presented.

2.1 Phase Transport Equations

The phase transport equations for the liquid and vapor phases can be expressed respectively as follows:

$$\frac{\partial(\rho_l \alpha_l)}{\partial t} + \nabla \cdot (\rho_l \alpha_l \boldsymbol{U_l}) = \dot{m} \tag{2.7}$$

$$\frac{\partial(\rho_v \alpha_v)}{\partial t} + \nabla \cdot (\rho_v \alpha_v \boldsymbol{U_v}) = -\dot{m} \tag{2.8}$$

On the LHS of the equations we have the temporal derivative and the convective term. Net source term is included on the RHS where $\dot{m} = \dot{m}_c - \dot{m}_e$. Phase transport equations do not include any diffusion term since the transport of volume fractions is only governed by the velocity. For conciseness, the two phase equations will be written as a single equation where the subscript 'i' denotes the phase.

$$\boxed{\frac{\partial(\rho_i \alpha_i)}{\partial t} + \nabla \cdot (\rho_i \alpha_i \boldsymbol{U_i}) = \pm \dot{m}} \tag{2.9}$$

It should be noted that in multiphase flows involving 'n' phases, 'n-1' phase transport equations are needed to completely solve for the phase distribution in the system. Since in this study two phases are involved, a single phase transport equation is sufficient. The model for determining the net rate of mass transfer will be discussed in detail in Chapter 3.

The implementation of the phase transport equation (liquid phase) already implemented in OpenFOAM is as following:

$$\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \boldsymbol{U}) + \nabla \cdot (\alpha_l \alpha_v \boldsymbol{U_r}) = \frac{\dot{m}_c + \dot{m}_e}{\rho_l} \tag{2.10}$$

It involves an addition of an extra artificial compression term to perform the compression of the interface. Here $\boldsymbol{U_r} = \boldsymbol{U_l} - \boldsymbol{U_v}$ is the field necessary to compress the interface. This term is only active in the interface region due to the presence of '$\alpha_l \alpha_v$' term. The magnitude of this artificial velocity $\boldsymbol{U_r}$ is governed by the compression factor *cAlpha* which is defined in the *fvSolution* file in the *system* directory of the OpenFOAM case file. A *cAlpha* value of '0' indicates no compression of the interface, whereas a value of '1' indicates conservative compression. Enhanced compression can also be achieved by defining a value greater than '1', however it is not recommended.

The starting point is the consideration of the velocity of the mixture as a weighted average.

$$\boldsymbol{U} = \alpha_l \boldsymbol{U_l} + \alpha_v \boldsymbol{U_v} \tag{2.11}$$

Starting from the general phase transport, Eq (2.9) we get:

$$\alpha_i \frac{\partial \rho_i}{\partial t} + \rho_i \frac{\partial \alpha_i}{\partial t} + \nabla \cdot (\rho_i \alpha_i \boldsymbol{U_i}) = \pm \dot{m} \tag{2.12}$$

The convection term can be expanded according to Eq (F.3) in Appendix (F) with $a = \rho_i$, $b = 1$, $c = \alpha_i$ and $\boldsymbol{V} = \boldsymbol{U_i}$.

$$\nabla \cdot (\rho_i \alpha_i \boldsymbol{U_i}) = \rho_i (\nabla \cdot (\alpha_i \boldsymbol{U_i})) + \alpha_i (\boldsymbol{U_i} \cdot \nabla \rho_i) \tag{2.13}$$

Inserting Eq (2.13) in Eq (2.12) we obtain:

$$\alpha_i \frac{\partial \rho_i}{\partial t} + \rho_i \frac{\partial \alpha_i}{\partial t} + \rho_i[\nabla\cdot(\alpha_i \boldsymbol{U_i})] + \alpha_i(\boldsymbol{U_i} \cdot \nabla \rho_i) = \pm \dot{m} \tag{2.14}$$

$$\rho_i \left[\frac{\partial \alpha_i}{\partial t} + \nabla \cdot (\alpha_i \boldsymbol{U_i})\right] = -\alpha_i \left[\frac{\partial \rho_i}{\partial t} + \boldsymbol{U_i} \cdot \nabla \rho_i\right] \pm \dot{m} \tag{2.15}$$

$$\frac{\partial \alpha_i}{\partial t} + \nabla \cdot (\alpha_i \boldsymbol{U_i}) = -\frac{\alpha_i}{\rho_i}\left[\frac{D\rho_i}{Dt}\right] \pm \frac{\dot{m}}{\rho_i} \tag{2.16}$$

For conciseness we will write the above equation as:

$$\boxed{\frac{\partial \alpha_i}{\partial t} + \nabla \cdot (\alpha_i \boldsymbol{U_i}) = -\frac{\alpha_i \dot{\rho}_i}{\rho_i} \pm \frac{\dot{m}}{\rho_i}} \tag{2.17}$$

From the above equation, liquid and vapor phase transport equations can be written as follows respectively.

$$\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \boldsymbol{U_l}) = -\frac{\alpha_l \dot{\rho}_l}{\rho_l} + \frac{\dot{m}}{\rho_l} \tag{2.18}$$

$$\frac{\partial \alpha_v}{\partial t} + \nabla \cdot (\alpha_v \boldsymbol{U_v}) = -\frac{\alpha_v \dot{\rho}_v}{\rho_v} - \frac{\dot{m}}{\rho_v} \tag{2.19}$$

Adding both the transport equations, Eq (2.18) & Eq (2.19), and utilizing Eq (2.11) leads to:

$$\frac{\partial(\alpha_l + \alpha_v)}{\partial t} + \nabla \cdot (\boldsymbol{U} - \alpha_v \boldsymbol{U_v}) + \nabla \cdot (\alpha_v \boldsymbol{U_v}) = -\frac{\alpha_l \dot{\rho}_l}{\rho_l} - \frac{\alpha_v \dot{\rho}_v}{\rho_v} + \dot{m}\left[\frac{1}{\rho_l} - \frac{1}{\rho_v}\right] \tag{2.20}$$

Finally we get the expression for divergence of the velocity field which is the term associated with the dilation of fluid particles. For incompressible flows, we know that this term is zero.

$$\boxed{\nabla \cdot \boldsymbol{U} = -\frac{\alpha_l \dot{\rho}_l}{\rho_l} - \frac{\alpha_v \dot{\rho}_v}{\rho_v} + \dot{m}\left[\frac{1}{\rho_l} - \frac{1}{\rho_v}\right]} \tag{2.21}$$

Now we consider Eq (2.18). Adding and subtracting $\nabla \cdot (\alpha_l \boldsymbol{U})$ on the LHS and utilizing Eq (2.11) leads to:

$$
\begin{aligned}
\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \boldsymbol{U}) + \nabla \cdot (\alpha_l \boldsymbol{U_l}) - \nabla \cdot (\alpha_l \boldsymbol{U}) &= -\frac{\alpha_l \dot{\rho}_l}{\rho_l} + \frac{\dot{m}}{\rho_l} \\
\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \boldsymbol{U}) + \nabla \cdot (\alpha_l \boldsymbol{U_l}) - \nabla \cdot [\alpha_l(\alpha_l \boldsymbol{U_l} + \alpha_v \boldsymbol{U_v})] &= -\frac{\alpha_l \dot{\rho}_l}{\rho_l} + \frac{\dot{m}}{\rho_l} \\
\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \boldsymbol{U}) + \nabla \cdot [\alpha_l \boldsymbol{U_l}(1-\alpha_l)] - \nabla \cdot [\alpha_l \alpha_v \boldsymbol{U_v}] &= -\frac{\alpha_l \dot{\rho}_l}{\rho_l} + \frac{\dot{m}}{\rho_l} \\
\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \boldsymbol{U}) + \nabla \cdot (\alpha_l \alpha_v (\boldsymbol{U_l} - \boldsymbol{U_v})) &= -\frac{\alpha_l \dot{\rho}_l}{\rho_l} + \frac{\dot{m}}{\rho_l}
\end{aligned}
\tag{2.22}
$$

So the modified liquid phase transport equation can now be written as:

$$
\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \boldsymbol{U}) + \nabla \cdot (\alpha_l \alpha_v \boldsymbol{U_r}) = -\frac{\alpha_l \dot{\rho}_l}{\rho_l} + \frac{\dot{m}}{\rho_l} \tag{2.23}
$$

Taking Eq (2.18) and adding and subtracting $\alpha_l(\nabla \cdot \boldsymbol{U})$ to the RHS of the equation we get:

$$
\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \boldsymbol{U_l}) = -\frac{\alpha_l \dot{\rho}_l}{\rho_l} + \alpha_l(\nabla \cdot \boldsymbol{U}) - \alpha_l(\nabla \cdot \boldsymbol{U}) + \frac{\dot{m}}{\rho_l} \tag{2.24}
$$

Substituting the expression for divergence of velocity, Eq (2.21), we obtain:

$$
\begin{aligned}
\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \boldsymbol{U_l}) &= -\frac{\alpha_l \dot{\rho}_l}{\rho_l} + \alpha_l[\nabla \cdot (\boldsymbol{U})] + \frac{\alpha_l^2 \dot{\rho}_l}{\rho_l} + \frac{\alpha_v \alpha_l \dot{\rho}_v}{\rho_v} \\
&\quad - \dot{m}\alpha_l \left[\frac{1}{\rho_l} - \frac{1}{\rho_v}\right] + \frac{\dot{m}}{\rho_l} \\
\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \boldsymbol{U_l}) &= -\frac{\alpha_l \dot{\rho}_l}{\rho_l}(1-\alpha_l) + \alpha_l[\nabla \cdot (\boldsymbol{U})] + \frac{\alpha_v \alpha_l \dot{\rho}_v}{\rho_v} \\
&\quad + \dot{m}\left[\frac{1}{\rho_l} - \alpha_l\left(\frac{1}{\rho_l} - \frac{1}{\rho_v}\right)\right] \\
\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \boldsymbol{U_l}) &= -\frac{\alpha_l \alpha_v \dot{\rho}_l}{\rho_l} + \frac{\alpha_v \alpha_l \dot{\rho}_v}{\rho_v} + \alpha_l[\nabla \cdot (\boldsymbol{U})] \\
&\quad + \dot{m}\left[\frac{1}{\rho_l} - \alpha_l\left(\frac{1}{\rho_l} - \frac{1}{\rho_v}\right)\right] \\
\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \boldsymbol{U_l}) &= \alpha_l \alpha_v \left[\frac{\dot{\rho}_v}{\rho_v} - \frac{\dot{\rho}_l}{\rho_l}\right] + \alpha_l[\nabla \cdot (\boldsymbol{U})] \\
&\quad + \dot{m}\left[\frac{1}{\rho_l} - \alpha_l\left(\frac{1}{\rho_l} - \frac{1}{\rho_v}\right)\right]
\end{aligned}
\tag{2.25}
$$

So we arrive at an expression for solving the liquid phase transport equation in OpenFOAM.

$$
\boxed{\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \boldsymbol{U_l}) = \alpha_l \alpha_v \left[\frac{\dot{\rho}_v}{\rho_v} - \frac{\dot{\rho}_l}{\rho_l}\right] + \alpha_l[\nabla \cdot (\boldsymbol{U})] + \dot{m}\left[\frac{1}{\rho_l} - \alpha_l\left(\frac{1}{\rho_l} - \frac{1}{\rho_v}\right)\right]} \tag{2.26}
$$

In OpenFOAM, the phase transport equation for the liquid phase is solved by the 'Multi Dimensional Universal Limiter with Explicit Solution' (MULES) to ensure boundedness of the liquid phase fraction. MULES is developed by Mr. Henry Weller and the method is not documented anywhere. Firstly the fluxes of the volume fraction are predicted from the velocity field and then an iterative correction is performed. This correction is done by limiting the phase fraction locally if it drops below zero or gets above unity in a specific cell.

2.2 Conservation of Mass

We can arrive at the expression for the conservation of mass for the whole system by replacing 'b' in (2.5) with ρ. We also consider that there are no diffusive fluxes for the transport of density ($\boldsymbol{F} = 0$). This can be expressed as:

$$\boxed{\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{U}) = 0} \tag{2.27}$$

The non conservative form can be written as:

$$\frac{\partial \rho}{\partial t} + \boldsymbol{U} \cdot \nabla \rho + \rho \nabla \cdot \boldsymbol{U} = 0 \tag{2.28}$$

So the divergence of velocity can be written as:

$$\nabla \cdot \boldsymbol{U} = -\frac{1}{\rho}\left[\frac{\partial \rho}{\partial t} + \boldsymbol{U} \cdot \nabla \rho\right] \tag{2.29}$$

$$\boxed{\nabla \cdot \boldsymbol{U} = -\frac{1}{\rho}\left[\frac{D\rho}{Dt}\right] = -\frac{\dot{\rho}}{\rho}} \tag{2.30}$$

2.3 Conservation of Linear Momentum

The conservation of linear momentum, also known as the Navier Stokes Equation, for any fluid can be expressed as:

$$\boxed{\frac{\partial \rho \boldsymbol{U}}{\partial t} + \nabla \cdot (\rho \boldsymbol{U}\boldsymbol{U}) = \rho \boldsymbol{g} + \nabla \cdot \boldsymbol{\sigma} + \sigma \kappa \nabla \alpha} \tag{2.31}$$

The curvature κ is defined as:

$$\kappa = -\nabla \cdot \frac{\nabla C}{|\nabla C|} \tag{2.32}$$

Due to the smoothing of the $\nabla\alpha$ field, the effects of the surface tension are only applied to the interface region which is spread over a few cells. Since, our problem deals with micrgravity, consideration of surface tension is the driving force for fluid motion [13].

A Newtonian fluid model is based on the following assumptions:

1. Shear stress is proportional to the rate of shearing strain.
2. Shear stress is '0' when the rate of shearing strain is '0'.
3. Stress to rate of strain relation is isotropic (no preferred direction in the fluid).

For a Newtonian Fluid we know that

$$\boldsymbol{\sigma} = -p_m \boldsymbol{I} + \mu[\nabla \boldsymbol{U} + (\nabla \boldsymbol{U})^T] + \lambda(\nabla \cdot \boldsymbol{U})\boldsymbol{I} \tag{2.33}$$

Assuming $\lambda = -\frac{2}{3}\mu$ we get:

$$\boldsymbol{\sigma} = -p_m \boldsymbol{I} + \mu[\nabla \boldsymbol{U} + (\nabla \boldsymbol{U})^T] - \frac{2\mu}{3}(\nabla \cdot \boldsymbol{U})\boldsymbol{I} \tag{2.34}$$

The mechanical pressure can be expressed in terms of the thermodynamic pressure and the bulk viscosity as follows:

$$p_m = p - \lambda(\nabla \cdot \boldsymbol{U}) \tag{2.35}$$

Inserting Eq (2.35) in Eq (2.34) we get:

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \mu[\nabla \boldsymbol{U} + (\nabla \boldsymbol{U})^T] + \left[\frac{2\mu}{3} + \lambda\right](\nabla \cdot \boldsymbol{U})\boldsymbol{I} \tag{2.36}$$

In most cases, the last term in the above equation can be neglected.

Substituting Eq (2.33) in the Eq (2.31) we obtain the Navier Stokes Equation for a Newtonian Fluid.

$$\begin{aligned} \frac{\partial \rho \boldsymbol{U}}{\partial t} + \nabla \cdot (\rho \boldsymbol{U}\boldsymbol{U}) &= \rho \boldsymbol{g} + \nabla \cdot \left[-p\boldsymbol{I} + \mu\left(\nabla \boldsymbol{U} + \nabla \boldsymbol{U}^T\right)\right] + \sigma\kappa\nabla\alpha \\ &= -\nabla p + \rho \boldsymbol{g} + \nabla \cdot \left[\mu\left(\nabla \boldsymbol{U} + \nabla \boldsymbol{U}^T\right)\right] + \sigma\kappa\nabla\alpha \end{aligned} \tag{2.37}$$

Pressure can also be written as:

$$p = p - \rho(\boldsymbol{g} \cdot \boldsymbol{h}) + \rho(\boldsymbol{g} \cdot \boldsymbol{h}) \tag{2.38}$$

$$p = p\_rgh + \rho(\boldsymbol{g} \cdot \boldsymbol{h}) \tag{2.39}$$

where p_rgh is the total pressure excluding the hydrostatic pressure. It is the variable which is solved for in OpenFOAM solvers.

An expression for the gradient of pressure can be derived as follows:

$$\nabla p = \nabla p\_rgh + \nabla(\rho \boldsymbol{g} \cdot \boldsymbol{h}) \tag{2.40}$$

$$\nabla p = \nabla p\_rgh + \rho \boldsymbol{g} + \boldsymbol{g} \cdot \boldsymbol{h} \nabla \rho \tag{2.41}$$

Substituting in Eq (2.37), we obtain the Navier Stokes Equation:

$$\boxed{\frac{\partial \rho \boldsymbol{U}}{\partial t} + \nabla \cdot (\rho \boldsymbol{U}\boldsymbol{U}) = -\nabla p\_rgh - \boldsymbol{g} \cdot \boldsymbol{h} \nabla \rho + \nabla \cdot \left[\mu \left(\nabla \boldsymbol{U} + \nabla \boldsymbol{U}^T\right)\right] + \sigma \kappa \nabla \alpha} \tag{2.42}$$

2.4 Conservation of Energy

The conservation of energy for the whole system can be expressed as:

$$\boxed{\frac{\partial(\rho e)}{\partial t} + \nabla \cdot (\rho e \boldsymbol{U}) = \nabla \cdot (\boldsymbol{\sigma} \cdot \boldsymbol{U}) - \nabla \cdot q} \tag{2.43}$$

Since the motion in the problem being studied is fairly low, the viscous dissipation term can be neglected in the energy equation.

It is generally considered that thermal equilibrium prevails over the interface, thereby giving rise to a continuous temperature field. A thermal non-equilibrium can only exist in cases of extreme transient cases. It is therefore practical to assume a continuous temperature distribution as we move from one phase to the other [14].

The energy equations for the liquid and vapor phases respectively can be expressed in terms of the energy density as follows:

$$\frac{\partial(\alpha_l \rho_l e_l)}{\partial t} + \nabla \cdot (\alpha_l \rho_l e_l \boldsymbol{U}) = \nabla \cdot (k_l \nabla T) + \dot{m} e_l \tag{2.44}$$

$$\frac{\partial(\alpha_v \rho_v e_v)}{\partial t} + \nabla \cdot (\alpha_v \rho_v e_v \boldsymbol{U}) = \nabla \cdot (k_v \nabla T) - \dot{m} e_v \tag{2.45}$$

They include the source term on the extreme right which deals with the amount of energy associated with one of the phases when phase change takes place. The above equations can be combined as follows:

$$\boxed{\frac{\partial(\alpha_i\rho_i e_i)}{\partial t} + \nabla\cdot(\alpha_i\rho_i e_i \boldsymbol{U}) = \nabla\cdot(k_i\nabla T) \pm \dot{m}e_i} \tag{2.46}$$

The specific energy comprises of the specific internal energy and the specific kinetic energy:

$$e = e_u + e_k \tag{2.47}$$

In this analysis we are going to neglect the effects of kinetic energy as the motion of the fluids is minimal.

$$e \approx e_u \tag{2.48}$$

The specific internal energy can be expressed in terms of the specific enthalpy and the latent heat of vaporization:

$$e = e_u = h - \frac{p}{\rho} + \mathit{\Delta} H_v \tag{2.49}$$

Substituting the expression of specific energy for phase 'i' in Eq (2.46) we get:

$$\begin{aligned}
\frac{\partial(\alpha_i\rho_i h_i)}{\partial t} + \nabla\cdot(\alpha_i\rho_i h_i\boldsymbol{U}) &= \left[\frac{\partial(\alpha_i p)}{\partial t} + \nabla\cdot(\alpha_i p\boldsymbol{U})\right] + k_i\nabla^2 T \\
&\quad \pm \dot{m}(h_i + \mathit{\Delta} H_v) \mp \dot{m}\left[\frac{p}{\rho_i}\right] \\
\frac{\partial(\alpha_i\rho_i c_{p_i} T)}{\partial t} + \nabla\cdot(\alpha_i\rho_i c_{p_i} T\boldsymbol{U}) &= \left[\frac{\partial(\alpha_i p)}{\partial t} + \nabla\cdot(\alpha_i p\boldsymbol{U})\right] + k_i\nabla^2 T \\
&\quad \pm \dot{m}(c_{p_i}T + \mathit{\Delta} H_v) \mp \dot{m}\left[\frac{p}{\rho_i}\right] \\
\frac{1}{\rho_i}\left[\frac{\partial(\alpha_i\rho_i T)}{\partial t} + \nabla\cdot(\alpha_i\rho_i T\boldsymbol{U})\right] &= \frac{1}{\rho_i c_{p_i}}\left[\frac{\partial(\alpha_i p)}{\partial t} + \nabla\cdot(\alpha_i p\boldsymbol{U})\right] + \frac{k_i}{\rho_i c_{p_i}}(\nabla^2 T) \\
&\quad \pm \frac{\dot{m}}{\rho_i}\left[T + \frac{\mathit{\Delta} H_v}{c_{p_i}}\right] \mp \dot{m}\left[\frac{p}{\rho_i^2 c_{p_i}}\right]
\end{aligned} \tag{2.50}$$

Expansion of the temporal derivative according to Eq (F.5) and the convection term according to Eq (F.3) in Appendix (F) yields:

$$\begin{aligned}
\alpha_i \left[\frac{\partial T}{\partial t} + \nabla\cdot(T\boldsymbol{U})\right] = {} & -\frac{\alpha_i T}{\rho_i}\left[\frac{\partial \rho_i}{\partial t} + (\boldsymbol{U}\cdot\nabla\rho_i)\right] - T\left[\frac{\partial\alpha_i}{\partial t} + (\boldsymbol{U}\cdot\nabla\alpha_i)\right] \\
& + \frac{\alpha_i}{\rho_i c_{p_i}}\left[\frac{\partial p}{\partial t} + \boldsymbol{U}\cdot\nabla p\right] + \frac{p}{\rho_i c_{p_i}}\left[\frac{\partial \alpha_i}{\partial t} + \boldsymbol{U}\cdot\nabla\alpha_i\right] \\
& + \frac{\alpha_i}{\rho_i c_{p_i}}\left[p(\nabla\cdot\boldsymbol{U})\right] + \frac{k_i}{\rho_i c_{p_i}}\left(\nabla^2 T\right) \\
& \pm \frac{\dot m}{\rho_i}\left[T + \frac{\Delta H_v}{c_{p_i}}\right] \mp \dot m\left[\frac{p}{\rho_i^2 c_{p_i}}\right]
\end{aligned} \tag{2.51}$$

Recognizing the material derivatives in the above equation, the modified phase energy equation can be written as follows:

$$\begin{aligned}
\alpha_i \left[\frac{\partial T}{\partial t} + \nabla\cdot(T\boldsymbol{U})\right] = {} & T\left[-\frac{\alpha_i\dot\rho_l}{\rho_i} \pm \frac{\dot m}{\rho_i}\right] - T\dot\alpha_i + \frac{p\dot\alpha_i}{\rho_i c_{p_i}} + \frac{\alpha_i}{\rho_i c_{p_i}}\left[\dot p + p(\nabla\cdot\boldsymbol{U})\right] \\
& + \frac{k_i}{\rho_i c_{p_i}}\left(\nabla^2 T\right) \pm \frac{\dot m \Delta H_v}{\rho_i c_{p_i}} \mp \dot m\left[\frac{p}{\rho_i^2 c_{p_i}}\right]
\end{aligned} \tag{2.52}$$

Adding the modified energy transport equations for liquid and vapor phase, Eq (2.52) we get:

$$\begin{aligned}
(\alpha_l + \alpha v)\left[\frac{\partial T}{\partial t} + \nabla\cdot(T\boldsymbol{U})\right] = {} & T\left[-\frac{\alpha_l\dot\rho_l}{\rho_l} - \frac{\alpha_v\dot\rho_v}{\rho_v} + \dot m\left(\frac{1}{\rho_l} - \frac{1}{\rho_v}\right)\right] - T(\dot\alpha_l + \dot\alpha_v) \\
& + \frac{p\dot\alpha_l}{\rho_l c_{p_l}} + \frac{p\dot\alpha_v}{\rho_v c_{p_v}} + \left[\frac{\alpha_l}{\rho_l c_{p_l}} + \frac{\alpha_v}{\rho_v c_{p_v}}\right]\left[\dot p + p(\nabla\cdot\boldsymbol{U})\right] \\
& + \frac{k_l}{\rho_l c_{p_l}}\left(\nabla^2 T\right) + \frac{k_v}{\rho_v c_{p_v}}\left(\nabla^2 T\right) + \dot m \Delta H_v\left[\frac{1}{\rho_l c_{p_l}} - \frac{1}{\rho_v c_{p_v}}\right] \\
& + \dot m p\left[\frac{1}{\rho_v^2 c_{p_v}} - \frac{1}{\rho_l^2 c_{p_l}}\right]
\end{aligned} \tag{2.53}$$

We also know that:

$$\begin{aligned}
\alpha_l + \alpha_v &= 1 \\
\frac{D\alpha_l}{Dt} + \frac{D\alpha_v}{Dt} &= 0
\end{aligned} \tag{2.54}$$

Utilizing the above expressions and the divergence of velocity Eq (2.21), we arrive at the energy equation for the mixture.

$$\begin{aligned}\left[\frac{\partial T}{\partial t} + \nabla \cdot (T\boldsymbol{U})\right] - T\left[\nabla \cdot \boldsymbol{U}\right] = \frac{p\dot{\alpha}_l}{\rho_l c_{p_l}} + \frac{p\dot{\alpha}_v}{\rho_v c_{p_v}} + \left[\frac{\alpha_l}{\rho_l c_{p_l}} + \frac{\alpha_v}{\rho_v c_{p_v}}\right]\left[\dot{p} + p(\nabla \cdot \boldsymbol{U})\right] \\ + \frac{k_l}{\rho_l c_{p_l}}\left(\nabla^2 T\right) + \frac{k_v}{\rho_v c_{p_v}}\left(\nabla^2 T\right) \\ + \dot{m}\mathit{\Delta} H_v\left[\frac{1}{\rho_l c_{p_l}} - \frac{1}{\rho_v c_{p_v}}\right] + \dot{m}p\left[\frac{1}{\rho_v^2 c_{p_v}} - \frac{1}{\rho_l^2 c_{p_l}}\right]\end{aligned} \tag{2.55}$$

2.5 Two Phase Properties

Since the model comprises of two different phases, there is a jump in properties as we move from one phase to the other. In numerical simulations, this jump needs to be modelled carefully over the interface which is spread over a few grid cells. The variation of properties in the two phase model are described and presented below:

$$\rho = \alpha_v\rho_v + \alpha_l\rho_l \tag{2.56}$$

$$\mu = \alpha_v\mu_v + \alpha_l\mu_l \tag{2.57}$$

$$c_p = \alpha_v c_{p_v} + \alpha_l c_{p_l} \tag{2.58}$$

$$k = \frac{k_v k_l}{k_v\alpha_l + k_l\alpha_v} \tag{2.59}$$

Some authors have used harmonic interpolation for the calculation of dynamic viscosity in the interface [3]. It should be noted that the above mentioned equations are valid for the whole domain. In the liquid phase region we have $\alpha_v = 0$ and in the vapor phase we have $\alpha_l = 0$.

Chapter 3

Interfacial Phase Change Phenomena

3.1 Mathematical Model

Evaporation and condensation processes make up the interfacial phase change model. The phenomena is shown in Figure 3.1.

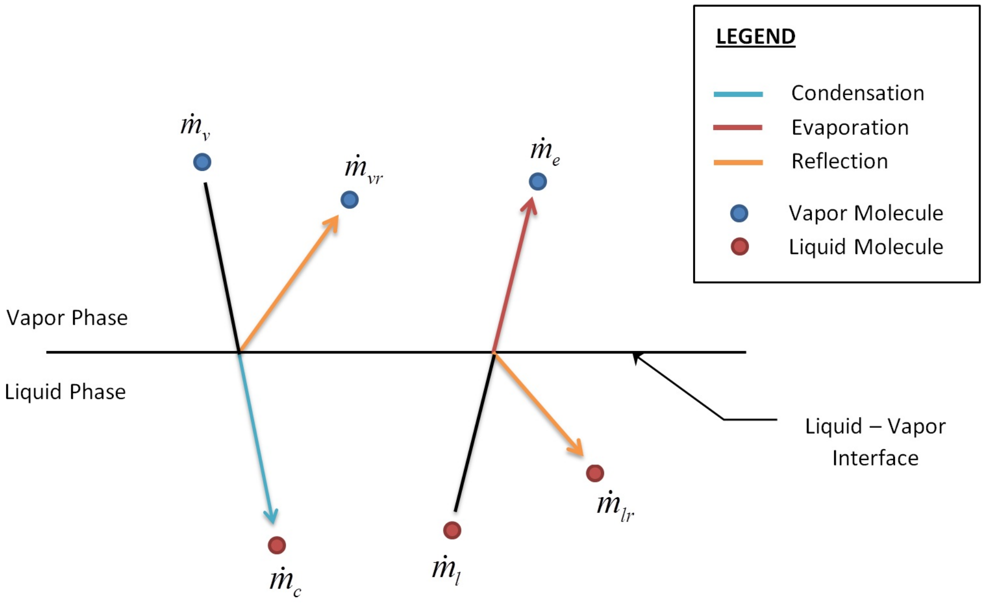


Figure 3.1: Evaporation and condensation process

The figure shows that the vapor molecules which strike the interface either enter the liquid phase by getting condensed or get reflected back into the vapor phase. On the other hand, when the liquid molecules strike the interface, some of them enter the vapor phase by evaporating and the rest get reflected back into the liquid phase. This is summarized in the equations below:

$$\begin{aligned} \dot{m}_v &= \dot{m}_c + \dot{m}_{vr} \\ \dot{m}_l &= \dot{m}_e + \dot{m}_{lr} \end{aligned} \tag{3.1}$$

The contributions made by Hertz and Knudsen to use the kinetic theory of gases have been vital to understand the evaporation and condensation phenomena [15]. The rate of evaporation and condensation can be described through the Kinetic Theory of Gases. According to it the rate of evaporation depends on the liquid pressure and temperature. Similarly the rate of condensation depends on vapor pressure and temperature. They proposed that the evaporation from the liquid surface of the Knudsen layer takes place according to a half range Maxwellian distribution. The Hertz-Knudsen equation can be stated as:

$$\dot{m} = \sqrt{\frac{M}{2\pi R_u}} \left[C_c \left(\frac{P_v}{\sqrt{T_v}} \right) - C_e \left(\frac{P_{sat}(T_l)}{\sqrt{T_l}} \right) \right] \tag{3.2}$$

In this formula, the accommodation coefficients for evaporation and condensation have been introduced. The accommodation coefficient for evaporation (C_e) represents the fraction of liquid molecules which get converted to vapor upon striking the liquid-vapor interface. Similarly, the accommodation coefficient for condensation (C_c) refers to the fraction of vapor molecules which get condensed upon striking the liquid-vapor interface. On the other hand, this relation proposed had a shortcoming that it did not account for the flow velocity which results due to the phase change on either side of the interface.

Another relation was proposed by Schrage [16] where the effect of the bulk vapor velocity was taken into account. For this purpose a surface S_i immediately adjacent to the interface, in the vapor region was considered. So when the gas moves normal to this planar surface with a speed of w_0, the flux of molecules passing through it in the direction of the bulk motion can be given as:

$$j_+ = \mathit{\Gamma}(a) \sqrt{\frac{M}{2\pi R_u T}} \left(\frac{P}{m} \right) \tag{3.3}$$

Similarly if molecules move in the direction opposite to that of the bulk motion then:

$$j_- = \mathit{\Gamma}(-a) \sqrt{\frac{M}{2\pi R_u T}} \left(\frac{P}{m} \right) \tag{3.4}$$

where $\mathit{\Gamma}$ is the distribution factor which corrects the effects of bulk gas motion and is given by the following relations:

$$\mathit{\Gamma}(a) = e^{a^2} + a\pi^{1/2}[1 + erf(a)] \tag{3.5}$$

$$\mathit{\Gamma}(-a) = e^{a^2} - a\pi^{1/2}[1 + erf(a)] \tag{3.6}$$

and a is defined as:

$$a = w_0 \sqrt{\frac{M}{2R_u T_v}} \tag{3.7}$$

Since the surface S_i is an infinitesimally small distance from the phase interface, it can be safely said that there is no bulk motion effect on molecules emerging from the liquid and passing through the surface.

After some calculations, the following expression for net mass flux rate through a liquid vapor interface is given by [16]:

$$\dot{m} = \sqrt{\frac{M}{2\pi R_u}} \left[C_c \left(\frac{\mathit{\Gamma} p_v}{\sqrt{T_v}} \right) - C_e \left(\frac{p_l}{\sqrt{T_l}} \right) \right] \tag{3.8}$$

The heat flux across the interface can be found by the product of net mass flux and the enthalpy of vaporization.

$$q_i = \mathit{\Delta} H_v \sqrt{\frac{M}{2\pi R_u}} \left[C_c \left(\frac{\mathit{\Gamma} p_v}{\sqrt{T_v}} \right) - C_e \left(\frac{p_l}{\sqrt{T_l}} \right) \right] \tag{3.9}$$

Since the rate of phase change is governed by the heat flux across the interface, the bulk velocity of the vapor is given by:

$$w_0 = \frac{q_i}{\rho_v \mathit{\Delta} H_v} \tag{3.10}$$

So a can be expanded to:

$$a = \frac{q_i}{\rho_v \mathit{\Delta} H_v} \sqrt{\frac{M}{2R_u T_v}} \tag{3.11}$$

In most cases this velocity is going to be extremely small and approximately equal to zero. In the case of $\lim\limits_{a \to 0}$ we can make the following approximations:

$$\begin{aligned} e^a &\approx 0 \\ erf(a) &\approx 0 \end{aligned} \tag{3.12}$$

So by utilizing assumptions (3.12) and Eq (3.11) the distribution factor can safely be assumed as follows:

$$\mathit{\Gamma} = 1 + a\pi^{1/2} \tag{3.13}$$

$$\mathit{\Gamma} = 1 + \frac{q_i}{\rho_v \mathit{\Delta} H_v} \sqrt{\frac{M\pi}{2R_u T_v}} \tag{3.14}$$

By expanding Eq (3.8) we get:

$$\begin{aligned}
\dot{m} &= \sqrt{\frac{M}{2\pi R_u}}\left[\left(1+\frac{q_i}{\rho_v \varDelta H_v}\sqrt{\frac{M\pi}{2R_uT_v}}\right)\frac{C_cp_v}{\sqrt{T_v}}-\frac{C_ep_l}{\sqrt{T_l}}\right]\\
\dot{m} &= \sqrt{\frac{M}{2\pi R_u}}\left[\frac{C_cp_v}{\sqrt{T_v}}+\frac{C_cq_i}{\varDelta H_v}\sqrt{\frac{M\pi}{2R_u}}\left(\frac{p_v}{\rho_vT_v}\right)-\frac{C_ep_l}{\sqrt{T_l}}\right]\\
\dot{m} &= \sqrt{\frac{M}{2\pi R_u}}\left[\frac{C_cp_v}{\sqrt{T_v}}+\frac{C_cq_i}{\varDelta H_v}\sqrt{\frac{M\pi}{2R_u}}\left(\frac{R_u}{M}\right)-\frac{C_ep_l}{\sqrt{T_l}}\right]\\
\dot{m} &= \sqrt{\frac{M}{2\pi R_u}}\left[\frac{C_cp_v}{\sqrt{T_v}}+\frac{C_cq_i}{\varDelta H_v}\sqrt{\frac{R_u\pi}{2M}}-\frac{C_ep_l}{\sqrt{T_l}}\right]
\end{aligned} \tag{3.15}$$

It should be noted that the 2nd term inside the square brackets from the previous equation is generally very small. This is due to the division by the enthalpy of vaporization and is therefore is frequently neglected in numerical analysis.

If the liquid and vapor phases are assumed to be in equilibrium, only then the accommodation coefficients for evaporation and condensation can be considered equal. This gives rise to equal rates of condensation and evaporation resulting in no net mass transfer. By assuming $C_c = C_e = C$ we obtain:

$$\begin{aligned}
\dot{m} &= C\sqrt{\frac{M}{2\pi R_u}}\left[\frac{p_v}{\sqrt{T_v}}+\frac{q_i}{\varDelta H_v}\sqrt{\frac{R_u\pi}{2M}}-\frac{p_l}{\sqrt{T_l}}\right]\\
\dot{m} &= C\sqrt{\frac{M}{2\pi R_u}}\left[\frac{p_v}{\sqrt{T_v}}-\frac{p_l}{\sqrt{T_l}}\right]+\frac{Cq_i}{\varDelta H_v}\sqrt{\frac{R_u\pi}{2M}}\sqrt{\frac{M}{2\pi R_u}}\\
\dot{m} &= C\sqrt{\frac{M}{2\pi R_u}}\left[\frac{p_v}{\sqrt{T_v}}-\frac{p_l}{\sqrt{T_l}}\right]+\frac{C}{2}\left(\frac{q_i}{\varDelta H_v}\right)\\
\dot{m}-\left(\frac{C}{2}\right)\dot{m} &= C\sqrt{\frac{M}{2\pi R_u}}\left[\frac{p_v}{\sqrt{T_v}}-\frac{p_l}{\sqrt{T_l}}\right]\\
\dot{m} &= \frac{2C}{2-C}\sqrt{\frac{M}{2\pi R_u}}\left[\frac{p_v}{\sqrt{T_v}}-\frac{p_l}{\sqrt{T_l}}\right]
\end{aligned} \tag{3.16}$$

This relation known as the Hertz-Knudsen-Schrage or Kucherov-Rickenglaz equation [16].

3.2 Implementation in OpenFOAM

In this section we are going to discuss the procedure adopted to develop a new solver *interEvapCondPhaseChangeFoam* in OpenFOAM capable of simulating the desired phase change phenomena.

3.2.1 Existing Solvers

The study began with the study of existing solvers already implemented in OpenFOAM. It was necessary to determine the capabilities of the existing solvers to set-up the starting point. Some preliminary simulations were also performed. These are listed below along with their short descriptions according to OpenFOAM Foundation [17].

1. *interFoam*
   “Solver for 2 incompressible, isothermal immiscible fluids using a VOF (volume of fluid) phase-fraction based interface capturing approach. The momentum and other fluid properties are of the ”mixture” and a single momentum equation is solved. Turbulence modelling is generic, i.e. laminar, RAS or LES may be selected.”

2. *compressibleInterFoam*
   “Solver for 2 compressible, non-isothermal immiscible fluids using a VOF (volume of fluid) phase-fraction based interface capturing approach. The momentum and other fluid properties are of the ”mixture” and a single momentum equation is solved. Turbulence modelling is generic, i.e. laminar, RAS or LES may be selected.”

3. *interPhaseChangeFoam*
   “Solver for 2 incompressible, isothermal immiscible fluids with phase-change (e.g. cavitation). Uses a VOF (volume of fluid) phase-fraction based interface capturing approach. The momentum and other fluid properties are of the ”mixture” and a single momentum equation is solved. The set of phase-change models provided are designed to simulate cavitation but other mechanisms of phase-change are supported within this solver framework. Turbulence modelling is generic, i.e. laminar, RAS or LES may be selected.”

3.2.2 Development of a New Solver

After the study of existing solvers it was decided to further develop the *interPhaseChangeFoam* solver. Since cavitation is not of interest for our problem several changes needed to be made in order to solve the problem correctly. Therefore, this solver was chosen as the starting point for the development of a new solver. The major inclusions which needed to be incorporated in 'interPhaseChangeFoam' are the following:

1. Energy Equation
   It is required to incorporate the energy equation to perform thermodynamic analysis and solve for the temperature distribution in the domain. It is mainly influenced by the non-isothermal boundary conditions and the initial temperatures of the phases.

2. Variable Fluid Properties
   Fluid properties such as density, viscosity, surface tension, thermal conductivity and specific heat capacity are greatly influenced by the variation in temperature. Therefore, it is necessary to include their dependence on temperature for such problems.

3. Interfacial Phase Change Equation
   The interfacial phase change equation (3.8) is incorporated in OpenFOAM in the file 'interfacialPhaseChangeModel.C'. It is done by introducing a couple of equations, one for modelling condensation and the other one for evaporation. The equations as implemented in OpenFOAM are given below:

   Condensation:
   C_c sqrt $\left(\frac{M}{2\pi RT}\right)$ max (P - P_{sat}, p0) * pos (0.09 - limitedAlpha1) * pos (limitedAlpha1 - 0.01) * pow (limitedAlpha1, 1);

   Evaporation:
   C_e sqrt $\left(\frac{M}{2\pi RT}\right)$ min (P - P_{sat}, p0) * pos (0.09 - limitedAlpha1) * pos (limitedAlpha1 - 0.01) * pow (scalar(1) - limitedAlpha1, 1);

The new solver is called *interEvapCondPhaseChangeFoam*.

Chapter 4

Benchmark Evaporation Case

4.1 Evaporation Model

The evaporation model considered here is based on the analytical study of one dimensional phase change of a vapor in contact with a heated wall by D. Jamet [19]. The model along with the boundary conditions is shown in Figure 4.1.

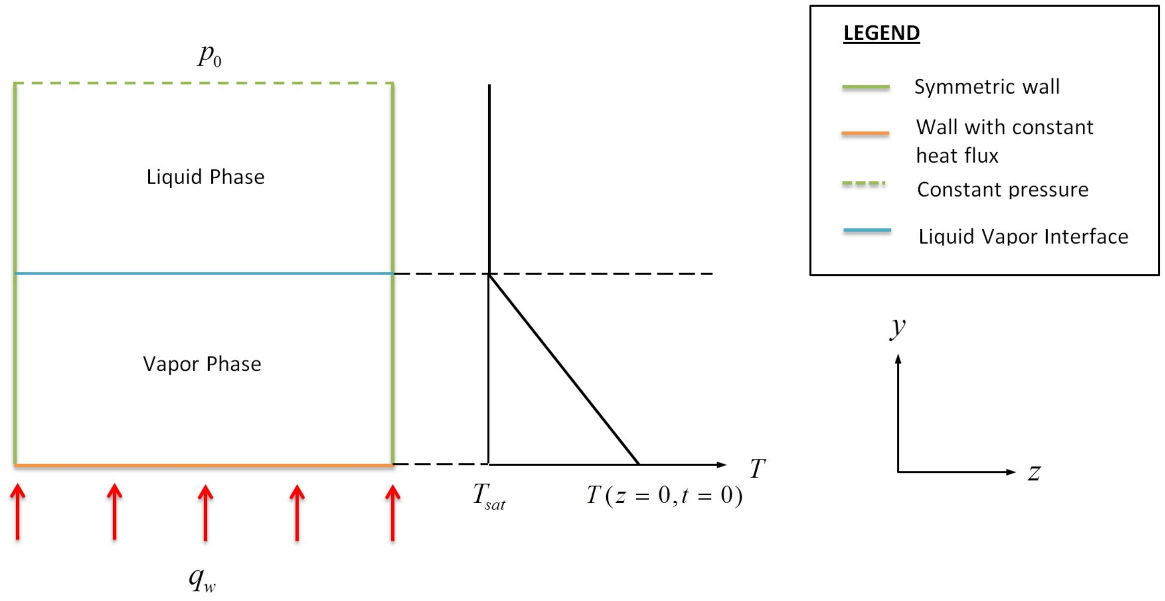


FIGURE 4.1: Benchmark evaporation model

The two phases in this model are assumed to be incompressible. Since the pressure at the upper boundary is fixed and equal to the saturation pressure, the system pressure is also the same. Therefore the interface is also at the saturation pressure. Additionally, due to the prevailing thermodynamic equilibrium condition at the interface, the interface temperature is equal to the saturation temperature.

The vapor is heated from the lower boundary by applying a wall heat flux of 6 Wm^{-2}. Slowly it's temperature begins to increase until it affects the interface. However, the interface temperature does not tend to increase since it is at saturation conditions. This means that the heat supplied from the lower boundary is absorbed at the interface and it helps in the vaporization process by keeping the temperature constant. This absorbed heat which is the cause of phase change is the Latent Heat of Vaporization or the Enthalpy of Vaporization.

The theoretical rate of evaporation can be expressed as [19]:

$$\dot{m}_e = \frac{q_w}{\Delta H_v} A_i \tag{4.1}$$

where A_i is the interface area and is equal to $1.65 \times 10^{-6}\ m^2$

The rate of evaporation calculated numerically based on KTG can be calculated by the Hertz-Knudsen-Schrage equation:

$$\dot{m}_e = C_e \sqrt{\frac{M}{2\pi R_u}} \left[\frac{p_l}{\sqrt{T_l}} \right] \tag{4.2}$$

4.2 Fluid Properties

The properties of the theoretical working fluid considered in this model are assumed to be constant and are mentioned in Table 4.1.

Property	Value	
Sigma	1x10^{-6}	Nm^{-1}
Contact Angle	90	degrees
Specific Gas Constant	333.3	$J/(kgK)$
Saturation Temperature @ 30 kpa	90	K

Table 4.1: General properties of benchmark fluid [20]

The fluid properties for both phases are constant and are presented in Table 4.2.

Property	Value	
	Liquid	Vapor
Density	1000	1
Specific Heat Capacity	2000	900
Thermal Conductivity	0.5	0.5

Table 4.2: Phase properties of benchmark fluid [20]

For the fluid in the benchmark case, we have $R = 333.3J/(kg\ K)$. We also know the saturation pressure and temperature of a point on the coexistence curve, $p_{sat} = 30\ kpa$ and $T_{sat} = 90\ K$.

The Clausius-Clapeyron Relation, derived in Appendix (E), after a few assumptions can be stated as follows:

$$\ln p = -\left(\frac{\Delta H_v}{R}\right)\frac{1}{T} + A \tag{4.3}$$

Applying it at the known saturation point we get the Clausius-Clapeyron Relation for the benchmark case in terms of the latent heat of vaporization as:

$$\ln p = \Delta H_v \left(3.33 \times 10^{-5} - \frac{3 \times 10^{-3}}{T}\right) + 10.309 \tag{4.4}$$

Evaluating for $\Delta H_v = 30\ kJ/kg$ yields:

$$p = \exp\left[-\frac{90}{T} + 11.308\right] \tag{4.5}$$

Evaluating for $\Delta H_v = 40\ kJ/kg$.

$$p = \exp\left[-\frac{120.012}{T} + 11.6424\right] \tag{4.6}$$

Evaluating for $\Delta H_v = 50\ kJ/kg$.

$$p = \exp\left[-\frac{150}{T} + 11.974\right] \tag{4.7}$$

4.3 Grid Independence Study (GIS)

Carrying out grid independence tests is vital in determining the precision of numerical results. It is used to describe an improvement of results by using successfully smaller grid sizes until they become grid independent. Seven grid sizes were studied for the evaporation model and their details are mentioned in Table 4.3.

Mesh No.	Number of Cells			
	x	y	z	Total
1	15	1	120	1800
2	25	1	200	5000
3	30	1	240	7200
4	35	1	280	9800
5	40	1	320	12800
6	45	1	360	16200
7	50	1	400	20000

Table 4.3: Grid sizes for GIS of evaporation benchmark model

The rates of evaporation per square meter were calculated for a test case with $C_e = 0.06$. The results are shown in Table 4.4 and Figure 4.2.

Mesh No.	Rate of Evaporation per m^2 ($\times10^{-5}$)	Percentage Difference (%)
1	19.11	—
2	11.29	-60.62
3	6.67	-78.8
4	9.04	26.26
5	8.12	-11.48
6	7.05	-15.04
7	6.68	-5.29

Table 4.4: Rates of evaporation for GIS ($C_e = 0.06$)

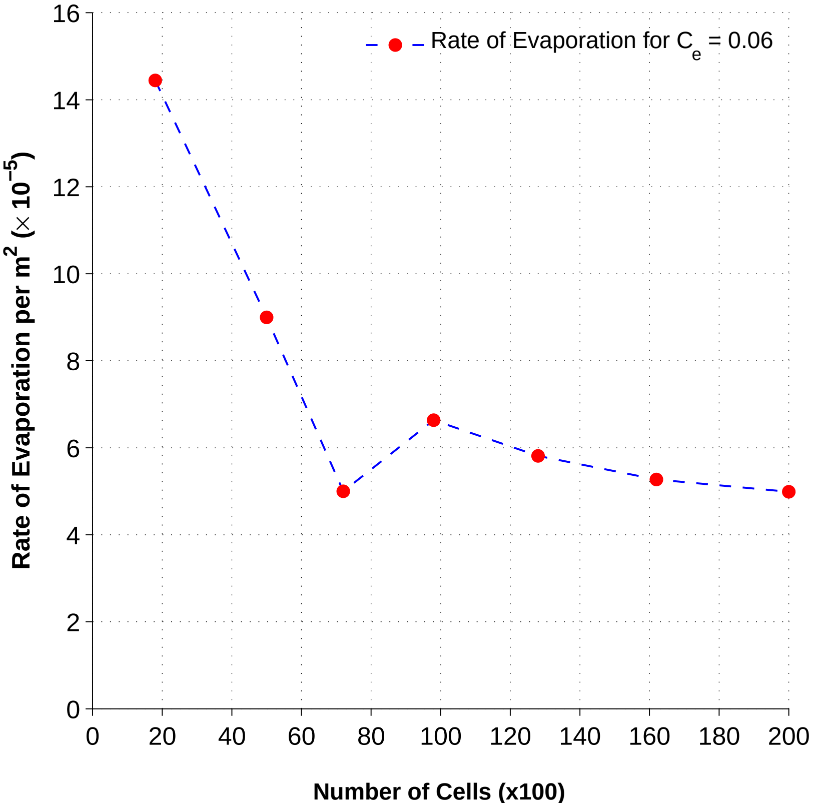


Figure 4.2: Rates of evaporation for GIS of BEC

We can easily deduce from the percentage difference mentioned in Table 4.4 and the Figure 4.2, that the difference between the results from Mesh No. 6 and 7 is significantly less. Therefore based on this GIS, it was decided to use the Mesh No. 6 as the reference mesh for further analysis.

4.4 Results and Discussion

In this section, the results obtained by simulating evaporation phenomena for the benchmark evaporation model are presented. The numerical results are compared with the theoretical ones for latent heats of vaporization of 30, 40 and 50 $KJ\ kg^{-1}$. The interface at the start of the simulation was considered to be at zero elevation.

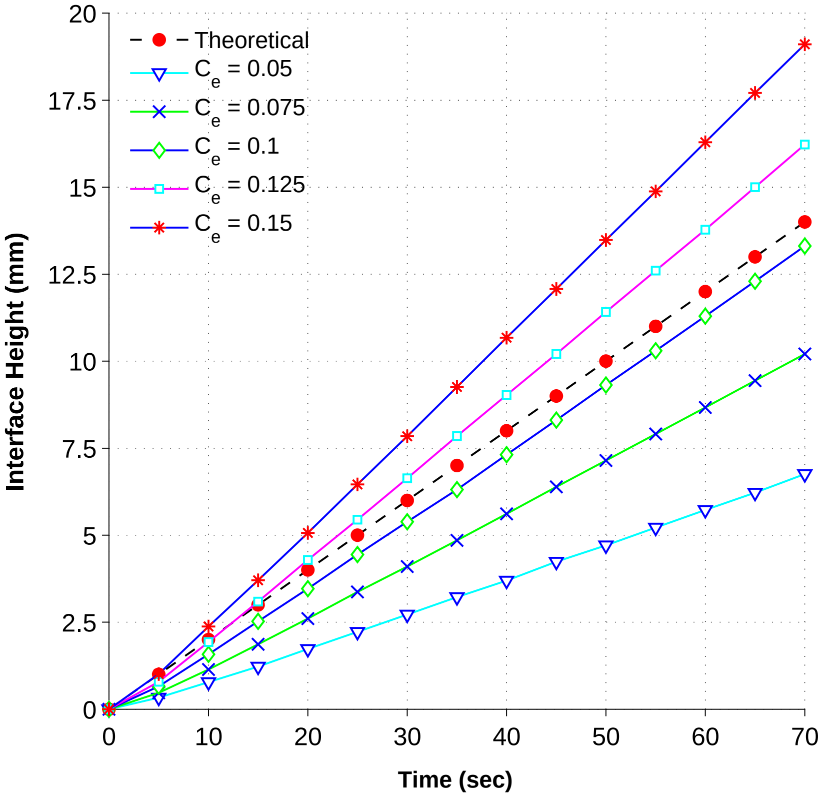


FIGURE 4.3: Interface height for BEC at $\Delta H_v = 30KJ/kg$

From the above graph it can be seen that an accommodation coefficient for evaporation in the range, $0.1 \leq C_e \leq 0.125$, will be able to complement the rate of evaporation as obtained through analytical solution.

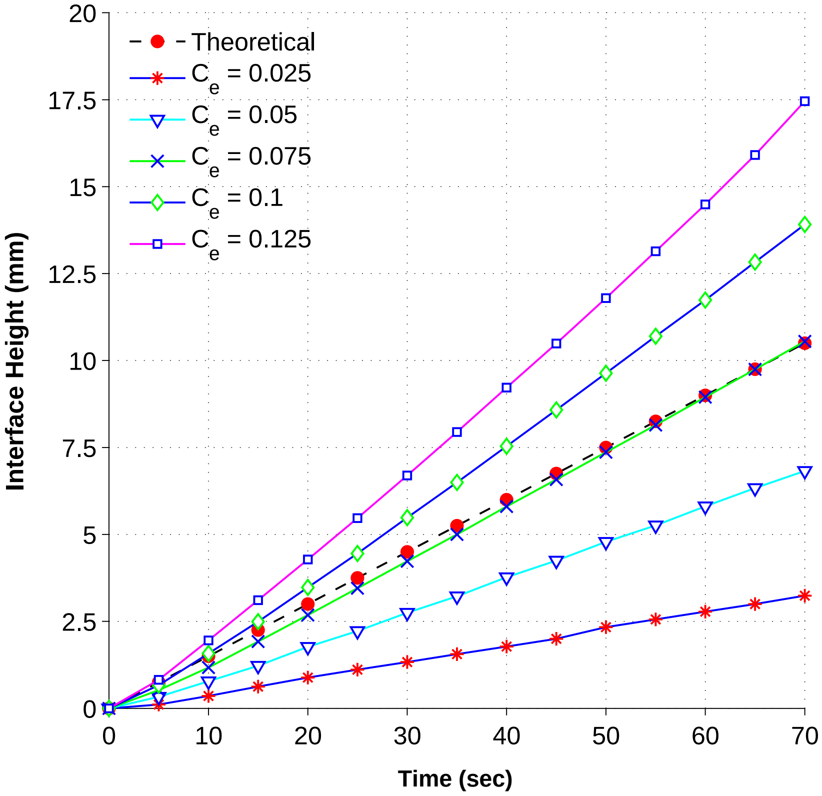


FIGURE 4.4: Interface height for BEC at $\Delta H_v = 40 KJ/kg$

Similarly for $\Delta H_v = 40\ KJ\ kg^{-1}$, we can see that C_e should be almost equal to 0.1.

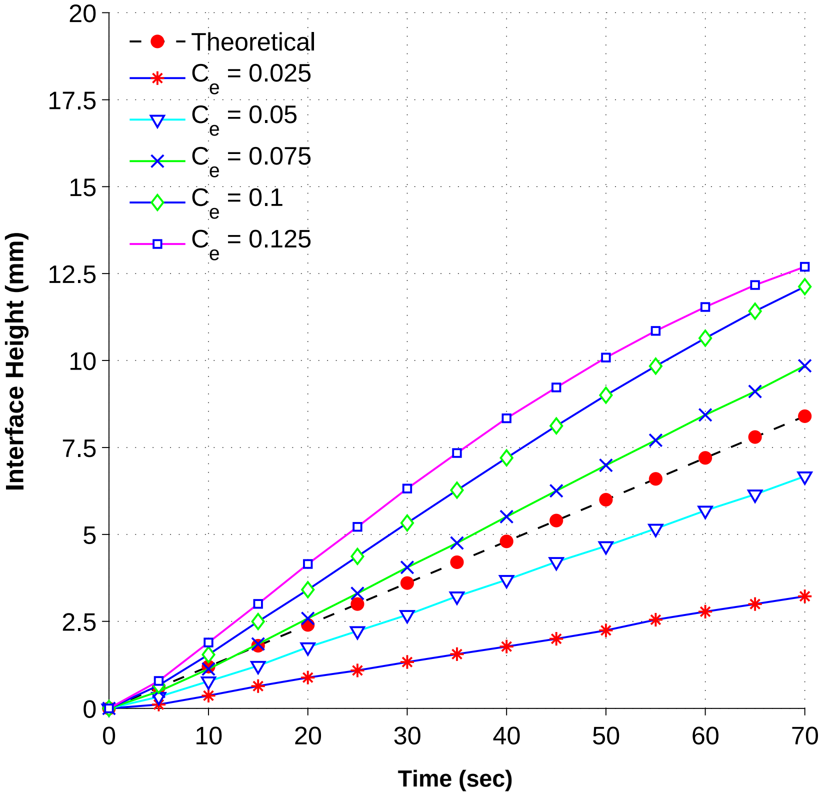


FIGURE 4.5: Interface height for BEC at $\Delta H_v = 50KJ/kg$

Finally for $\Delta H_v = 50\ KJ\ kg^{-1}$, theoretical rate of evaporation can be achieved by using an accommodation coefficient for evaporation in the range, $0.05 \leq C_e \leq 0.075$.

The above figures can be summarized in Figure 4.6.

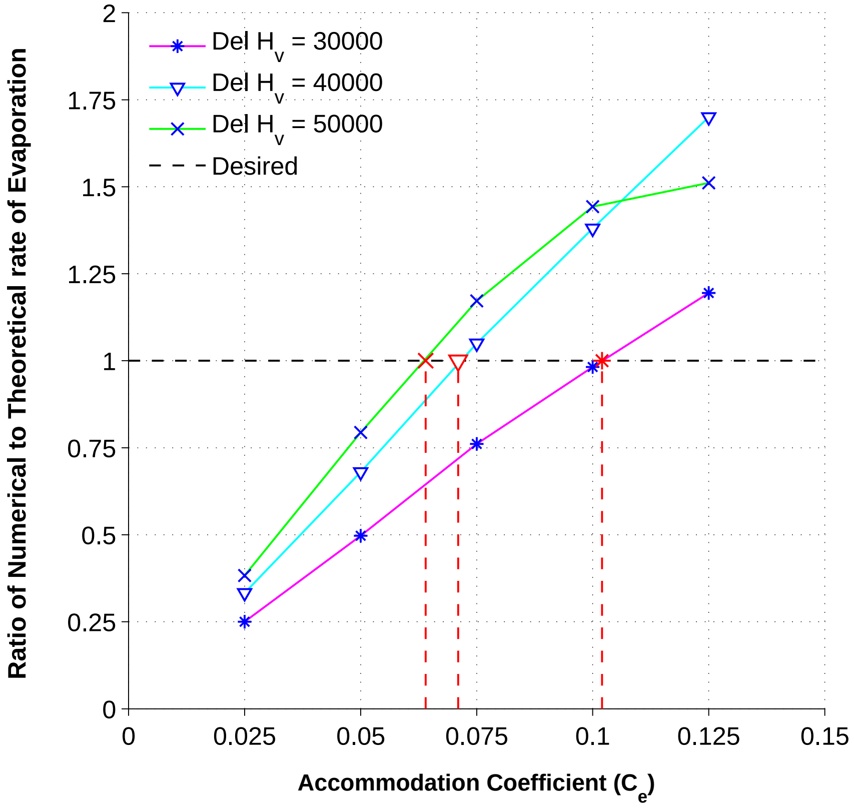


FIGURE 4.6: Comparison of numerical and theoretical evaporation rates for benchmark case

From the above graph we can get a reasonable estimate of the accommodation coefficient for evaporation that is required to achieve the theoretical mass transfer rate. This can be done by extrapolating a vertical line (red dotted) from the point of intersections (red markers) of the three continuous lines with the dotted desired line and the horizontal axis. The data extracted from Figure 4.6 can be summarized in the Table 4.5 and Figure 4.7.

ΔH_v $(KJ\ kg^{-1})$	C_e
30	0.102
40	0.071
50	0.064

TABLE 4.5: Actual C_e for BEC

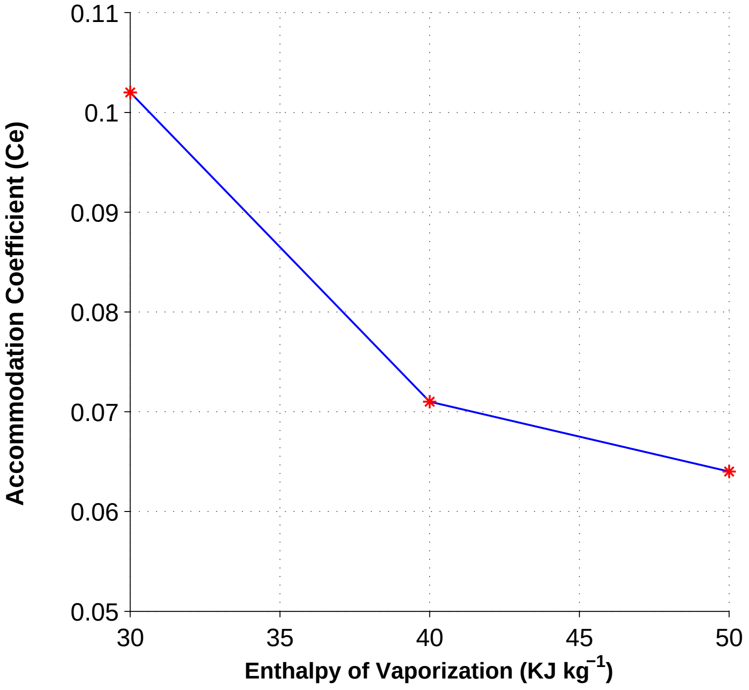


Figure 4.7: Actual C_e for BEC

So as we can observe from the graph above that as the enthalpy of vaporization is increased, the accommodation coefficient decreases. This is consistent with the theory since increasing the enthalpy of vaporization would mean that now more energy is required by the liquid molecules to escape the liquid surface and convert to vapor. Therefore the rate of evaporation would be relatively lower and so shall be the accommodation coefficient for evaporation. Extending the analysis further would reveal that the slope of the graph continues to decrease.

Chapter 5

Sounding Rocket Compere Experiment 2

5.1 Experimental Details

The German-French collaborative program COMPERE, focuses on the study of the behaviour of propellants in launcher tanks. SOURCE 2 is the second in series of small scale experiments performed under this banner. The sounding rocket MASER 12 which carried the SOURCE 2 test cell, was launched on 13th February 2012 from Esrange Space Center near Kiruna, Sweden. It was a total of six minute flight attaining a maximum altitude of 260 kilometers in microgravity. The experiment comprises of several phases but for the current study the first two phases are considered.

The void SOURCE 2 test cell before the start of the experiment is shown in Figure 5.1

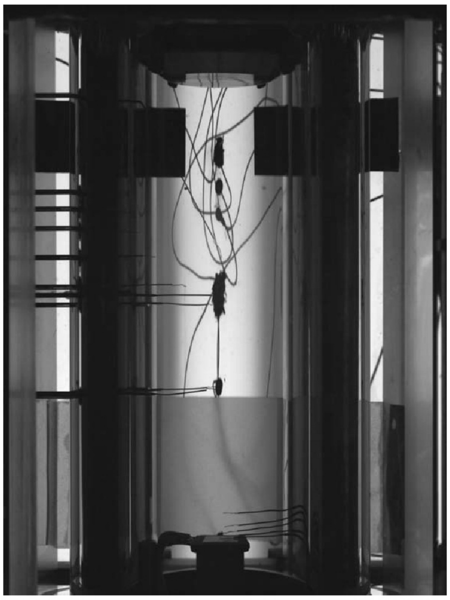


Figure 5.1: SOURCE 2 test cell at the start of experiment

5.1.1 First Phase

The first phase comprises of the filling of HFE-7000 vapor and HFE-7000 liquid into the SOURCE 2 test cell. The first phase has a time span of 20 seconds. Superheated vapor (403 K) is injected from the top inlet for the first 10 seconds. Simultaneously sub-cooled liquid is entered from the bottom of tank r1 for first 20 seconds of the experiment. The sequence during the filling process is shown below.

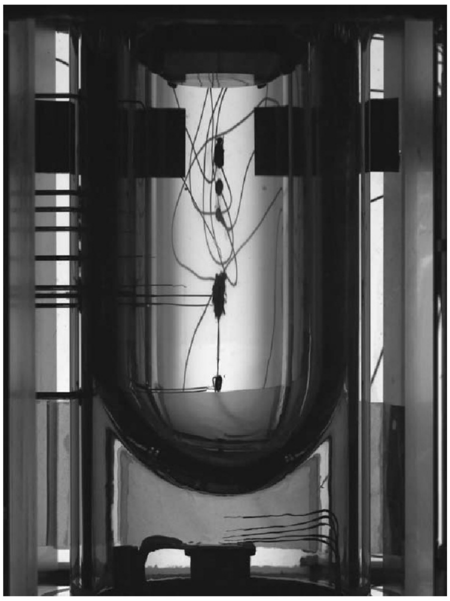


Figure 5.2: SOURCE 2 test cell at t = 13 seconds

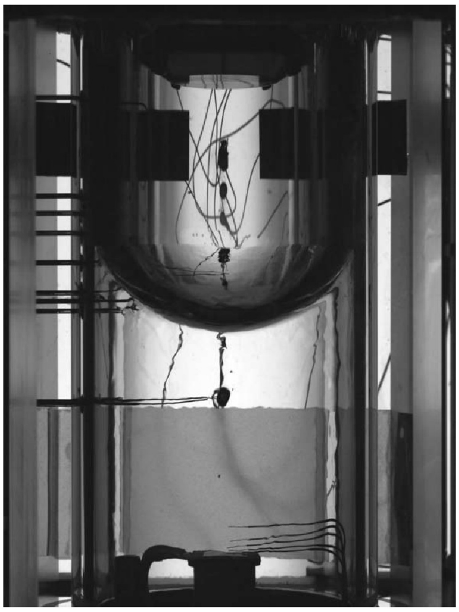


Figure 5.3: SOURCE 2 test cell at t = 20 seconds

The curved liquid-vapor interface can clearly be seen in Figures 5.2 and 5.3. Since the experiment is being conducted in microgravity conditions, the effect of surface tension force on the liquid-vapor interface is much more dominant than the gravitational force. This results in an interface which takes a half spherical shape and completely wets the tank r1 walls.

5.1.2 Second Phase

The time span of the second phase is 22 seconds. At the start of this phase both the inlet ports are shut and we have a closed system throughout this phase. As the hot vapor comes in contact with the sub-cooled liquid heat transfer takes place resulting in the cooling of the vapor and heating of the liquid in the interface region. Due to this heat exchange we expect the interface at some point to achieve the saturation conditions and the evaporation and condensation phenomena starts taking effect.

During the experiment it was observed that the condensation phenomena dominates and there is an increase in the liquid volume. The test cell at the end of the second phase is shown in Figure 5.4.

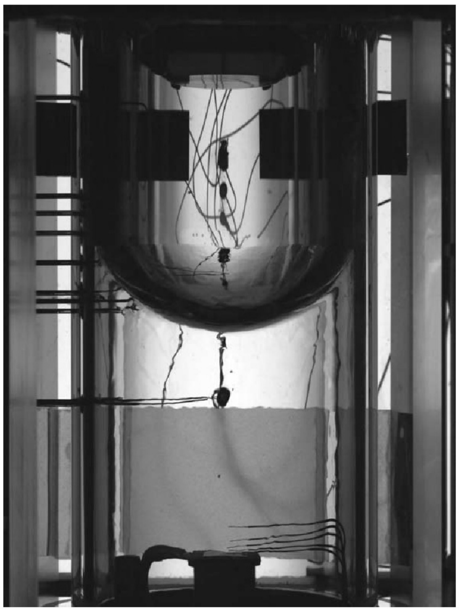


Figure 5.4: SOURCE 2 test cell at t = 42 seconds

Chapter 6

Numerical Study

The methodology adopted to perform the numerical study once the solver was developed can be summarised in the following flowchart.

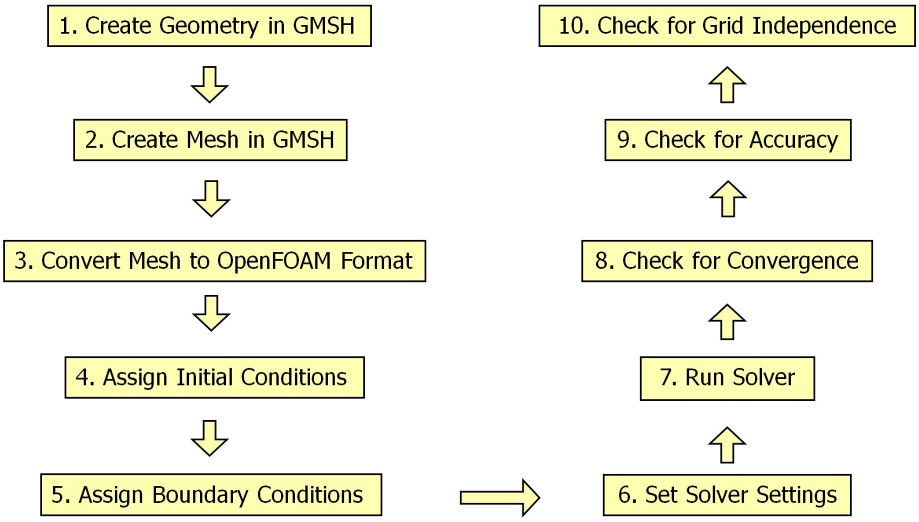

1. Create Geometry in GMSH
2. Create Mesh in GMSH
3. Convert Mesh to OpenFOAM Format
4. Assign Initial Conditions
5. Assign Boundary Conditions
6. Set Solver Settings
7. Run Solver
8. Check for Convergence
9. Check for Accuracy
10. Check for Grid Independence

FIGURE 6.1: Flowchart summarising numerical methodology

6.1 Geometry

The SOURCE 2 geometry can basically be divided into four components. These are:

1. Top component
2. Bottom component
3. Tank r1
4. Gas Port Phase Separator (GPPS)

Tank r1 is a cylindrical tank made up of glass for easy visualization of the fluid behaviour during the experiment.

The SOURCE 2 geometry along with the dimensions is shown in Figure 6.2. In order to reduce the simulation time, a 2 dimensional numerical study has been performed by defining an axi-symmetric geometry. The geometry as modelled in Gmsh is shown in Figure 6.3. The 2 mm long cylindrical section for vapor inlet has not been modelled since it is too narrow for numerical simulations. Instead, an inlet is defined just at the top of the GPPS.

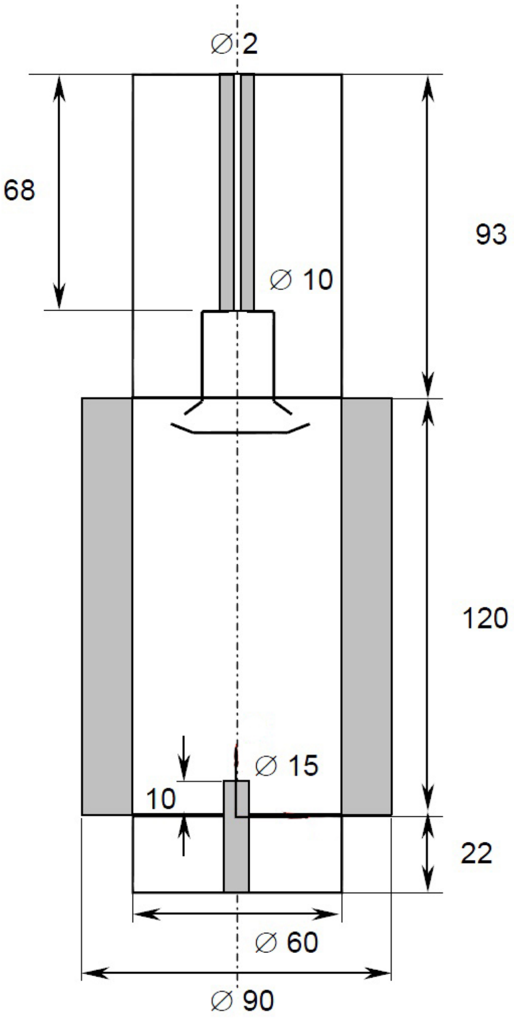


FIGURE 6.2: SOURCE 2 geometry with dimensions

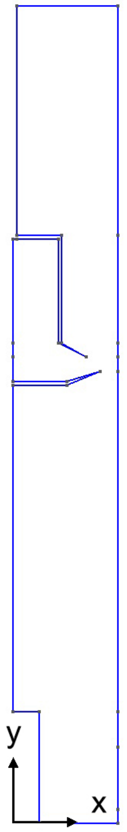


FIGURE 6.3: Axi-symmetric SOURCE 2 geometry as modelled in Gmsh

6.2 Meshing

The meshing was also performed in Gmsh and a pseudo three dimensional structured mesh was created. The mesh is shown in Figures 6.4 and 6.5.

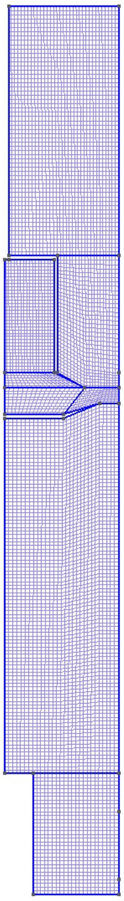


FIGURE 6.4: SOURCE 2 test cell meshing in Gmsh

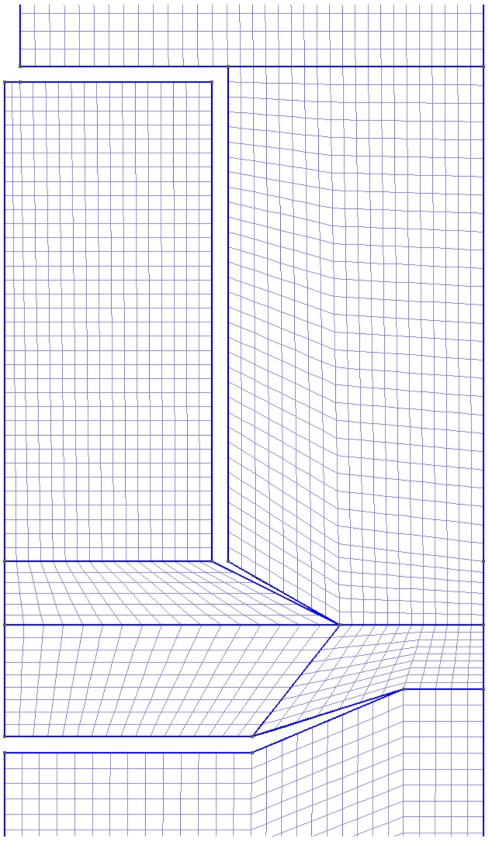


FIGURE 6.5: SOURCE 2 test cell meshing in the vicinity of GPPS

It must be stated that creating a three dimensional mesh is vital because OpenFOAM only reads three dimensional meshes. In order to convert the mesh from Gmsh format to OpenFOAM format, **gmshToFoam** command was used.

6.3 Initial Conditions

6.3.1 Tank r1

The initial pressure and the initial mean temperature in tank r1 is given below:

$$p_{r1} = 78\ kpa \tag{6.1}$$

$$T_{r1}(mean) = 358\ K \tag{6.2}$$

6.4 Boundary Conditions

6.4.1 Walls

A no slip boundary condition was applied to all the walls of the SOURCE 2 geometry. A static contact angle of zero degrees was specified to all the vertical walls.

(a) Top Component

The top component fully occupied by HFE-7000 vapor through out the experiment and its walls are maintained at a constant temperature of 418 K.

(b) Bottom Component

The walls of the bottom component are maintained at a constant temperature of 308 K.

(c) Tank r1

The variation of temperature specified on this wall is given by the following relation:

$$T_{r1w} = 294.28 + 12.56\ \exp^{((y-0.022)/0.05212)} \tag{6.3}$$

where the y is height of the source 2 test cell. The above temperature distribution is valid for the range $0.022 \geq z \leq 0.142$ meters.

(d) GPPS

The temperature of the GPPS is fixed at 423 K.

6.4.2 Inlets

(a) Liquid Inlet

There is an inlet at the bottom of approximately 1.5×10^{-5} m^2 for pumping in HFE-7000 liquid for the first 20 seconds of the experiment. The volume flow rate and temperature at the bottom inlet boundary are listed below:

$$\begin{aligned} \dot{V}_{bib} &= 1.18 \times 10^{-5}\ m^3\ s^{-1} \\ T_{bib} &= 299\ K \end{aligned} \tag{6.4}$$

(b) Vapor Inlet

HFE 7000 vapor is used to pressurize the tank r1 through the top inlet boundary for the first 10 seconds of the experiment. The inlet area for the actual geometry is 2×10^{-6} m^2. The volume flow rate and temperature of HFE-7000 vapor during pressurization is mentioned below:

$$\begin{aligned} \dot{m}_{tib} &= 0.78\ g\ s^{-1} \\ T_{tib} &= 403\ K \end{aligned} \tag{6.5}$$

6.4.3 Axis of Symmetry

The central plane which is the axis of rotation of the 2 dimensional SOURCE 2 test cell is specified with a *symmetryPlane* boundary condition. This implies that there is a mirror image of fluid flow on the other side of the axis by specifying a value of zero to the component of the gradient normal to the plane for every quantity. It also suggests that there is no flow across this boundary.

6.4.4 Summary

A summary of the types of boundary conditions as imposed in OpenFOAM is shown in Table 6.1.

Patches	Boundary Conditions			
	α_1	T	$\boldsymbol{U}$	*p_rgh*
1. Inlets				
Vapor	*inletOutlet*	*fixedValue*	*flowRateInletVelocity*	*fixedValue*
Liquid	*inletOutlet*	*fixedValue*	*flowRateInletVelocity*	*zeroGradient*
2. Walls				
Top	*zeroGradient*	*fixedValue*	*fixedValue*	*fixedFluxPressure*
Vertical	*constantAlphaContactAngle*	*fixedValue*	*fixedValue*	*fixedFluxPressure*
Bottom	*zeroGradient*	*fixedValue*	*fixedValue*	*buoyantPressure*
3. Axis	*symmetryPlane*	*symmetryPlane*	*symmetryPlane*	*symmetryPlane*

Table 6.1: Types of BCs implemented in OpenFOAM for SOURCE 2

6.5 Working Fluid

HFE 7000 ($C_3F_7OCH_3$), a complex organic solvent, is the working fluid in this study and its properties (density, viscosity, specific heat etc.) are defined by the relations mentioned in Appendix C. Due to time restrictions the enthalpy of vaporization of HFE-7000 was taken to be constant at 100 KJ kg^{-1}. In reality the enthalpy of vaporization is a function of temperature and should be implemented as a variable. It is due to its known and appropriate properties that it is chosen as the working fluid for the experiment. Its relatively low boiling point ensures that HFE 7000 liquid can be pumped at very low temperatures. It is also a non-flammable, low toxic and non corrosive fluid. In addition it is also a greenhouse gas.

The properties of HFE 7000 at 25 deg C & 1 atm (RTP) are mentioned in Table 6.2.

Property	Value		
	Liquid	Vapor	Units
Density	1400	8.2	kgm^{-3}
Kinematic Viscosity	3.0×10^{-7}	3.9×10^{-7}	m^2s^{-1}
Specific Heat Capacity	1300	847.8	$Jkg^{-1}K^{-1}$
Thermal Conductivity	0.075	0.083	$Wm^{-1}K^{-1}$

Table 6.2: Phase properties of HFE 7000 at RTP

6.6 Pressure Velocity Coupling

Generally in CFD, the pressure velocity coupling is mostly done by the PISO method, SIMPLE method and their derivatives. PISO stands for 'Pressure Implicit with Splitting of Operators' and SIMPLE for 'Semi Implicit Method for Pressure Linked Equations'.

The pressure velocity coupling available in OpenFOAM is described in [10]. Since the present study deals with the study of a transient problem, PISO algorithm is used.

6.7 Settings

The time step size in the simulations performed was determined by the Courant Number which determines the propagation speed of information on the mesh. The Courant Number is defined as follows:

$$Co = \frac{u\Delta t}{\Delta x} \tag{6.6}$$

It gives a measure of the distance a fluid particle travels in a single time step compared to the size of the cell. The upper limit for Courant number is unity, which will give the largest time step. A Courant Number greater than unity would imply that the information about fluid motion is being passed too rapidly from one mesh cell to the other, leading to erroneous results. In this study a maximum Courant Number of 0.5 was considered to be on the safe side.

6.8 Discetization Procedures

Discretization refers to the process of breaking a continuous problem into discrete quantities. In general discretizations can be subdivided into the following categories [21]

Spatial Discretization

It deals with discretization of the solution domain by a set of points that fill and bound the region of space.

Temporal Discretization

Temporal Discretization is necessary for transient problems and defines how the time domain needs to be divided into finite number of time steps. Euler implicit, explicit and Crank Nicholson schemes can be specified.

Equation Discretization

Equation discretization deals with the procedure to discretize the partial differential Equation that governs the problem. It involves the generation of a system of algebraic equations in terms of discrete quantities which are defined at specific points in the domain.

Chapter 7

Results and Discussion

The results obtained from numerical simulations are presented in this section. Three cases were solved to determine the effect of varying the accommodation coefficients on the phase change phenomena. The details of the cases are presented in Table 7.1.

Case No.	Accommodation Coefficients	
	Evaporation	Condensation
1	0.05	0.1
2	0.1	0.1
3	0.2	0.1

Table 7.1: Accommodation coefficients for numerical simulations

The initial guess values chosen for the accommodation coefficients were based on the experimental studies carried out by researchers previously. Due to time restrictions it was decided to keep a constant accommodation coefficient for condensation ($C_e = 0.1$) and vary the accommodation coefficient for evaporation to have an outlook at the net mass transfer. For a better understanding of the phase change process, different combinations of the accommodation coefficients should be utilized in future.

7.1 Phase Distribution

The results for the distribution of the liquid and vapor phases is presented in this section. The interface height during the filling process is presented in Figure 7.1 and is compared with the experimental results.

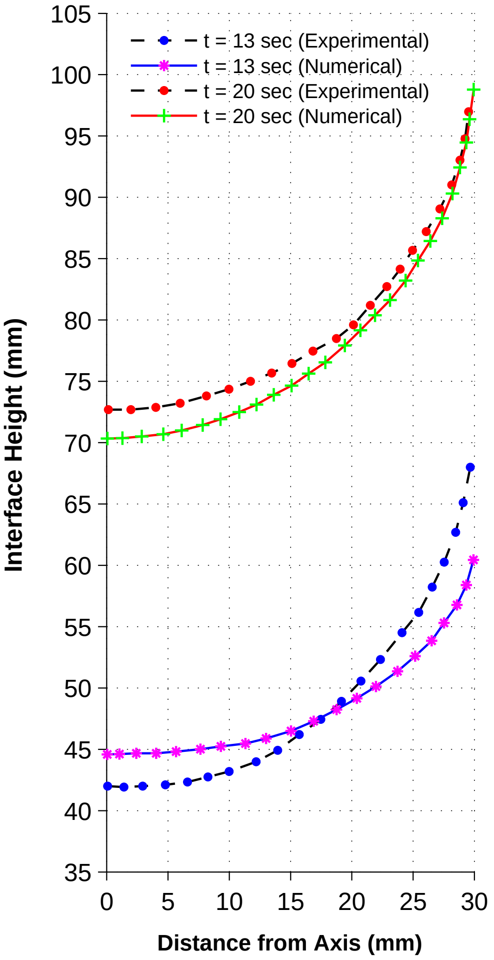


Figure 7.1: Comparison of experimental and numerical interface position during the 1st phase

Here we can see that the interface heights calculated numerically are pretty much in agreement with the experimental results. Here it must be mentioned that the interface height was not measured during the experiment. However, since a couple of photographs of the SOURCE 2 test cell were captured, a rough estimate can be obtained. The estimate would be quite accurate along the central plane but as we move towards the wall, it will tend to deviate from the actual value. The reason for this lies in the fact that the quartz glass cell has a square cross-section while the cavity has a cylindrical cross-section giving rise to refraction phenomena. Therefore, the interface height would be slightly underestimated from the photographs and this underestimation would be zero at the central plane and maximum at the wall. The deviation between numerical and experimental results could be accounted to the fact that the filling rates during the experiment were observed to be varying, however a constant filling rate was used for the numerical simulations.

The interface position during the 2nd phase as simulated for case 1 is shown in Figure 7.2.

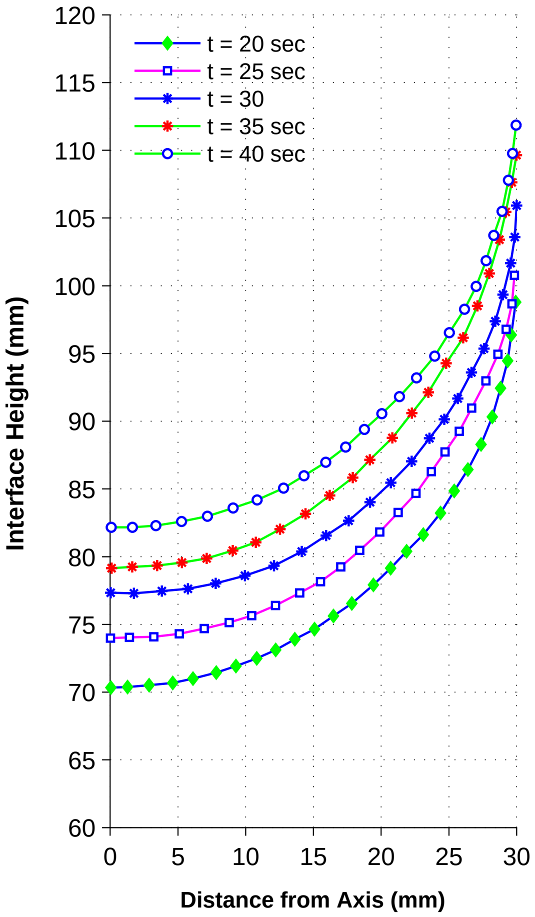


FIGURE 7.2: Interface position obtained numerically during 2nd phase for case 1

7.2 Temperature Distribution

At the start of the experiment we have a stratified temperature distribution due to the fixed temperature boundary conditions specified on the SOURCE 2 test cell boundaries. The test cell is just filled with vapor at this moment. Figure 7.3 shows the temperature distribution at the beginning of the first phase.

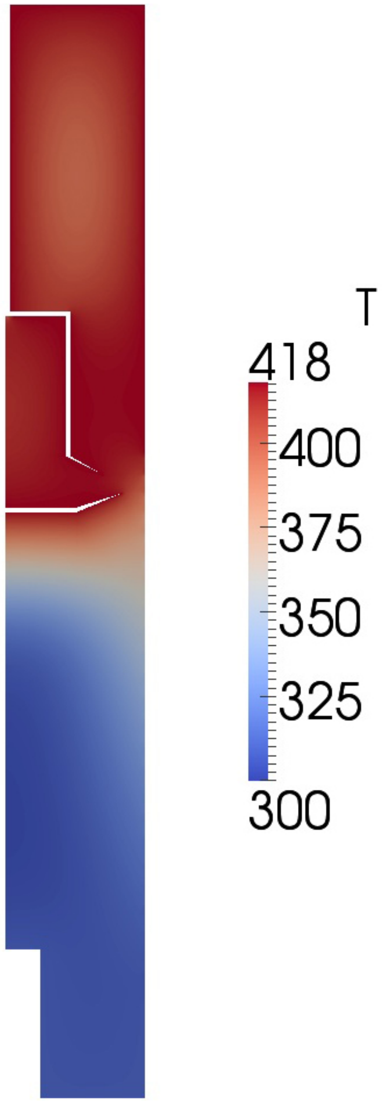


FIGURE 7.3: Temperature contours at 0 seconds for case 1

Temperature contours during the first phase are shown in Figure 7.4. The interface position is marked by a black line.

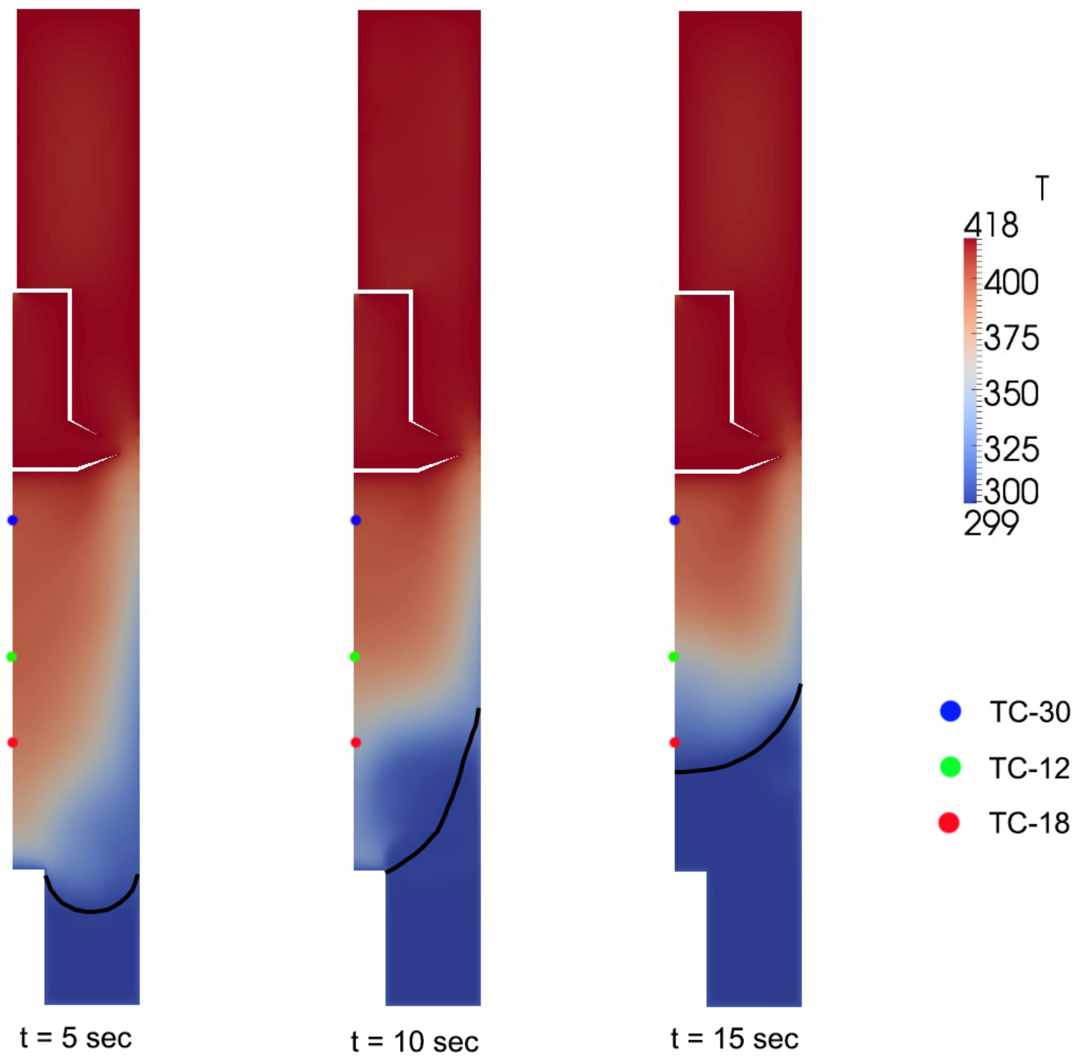


Figure 7.4: Temperature contours at 0, 10 and 15 seconds for case 1

We can see that the interface position rises as relatively cold liquid is pumped in from the bottom inlet boundary shown by the blue region at the bottom of the cell. At the same time hot vapor is injected from the top boundary of GPPS, thereby increasing the average temperature of the vapor which is shown by the red region. Near the interface heat exchange takes place between the cold liquid and hot vapor.

The temperature contours during the second phase are shown in the figures below:

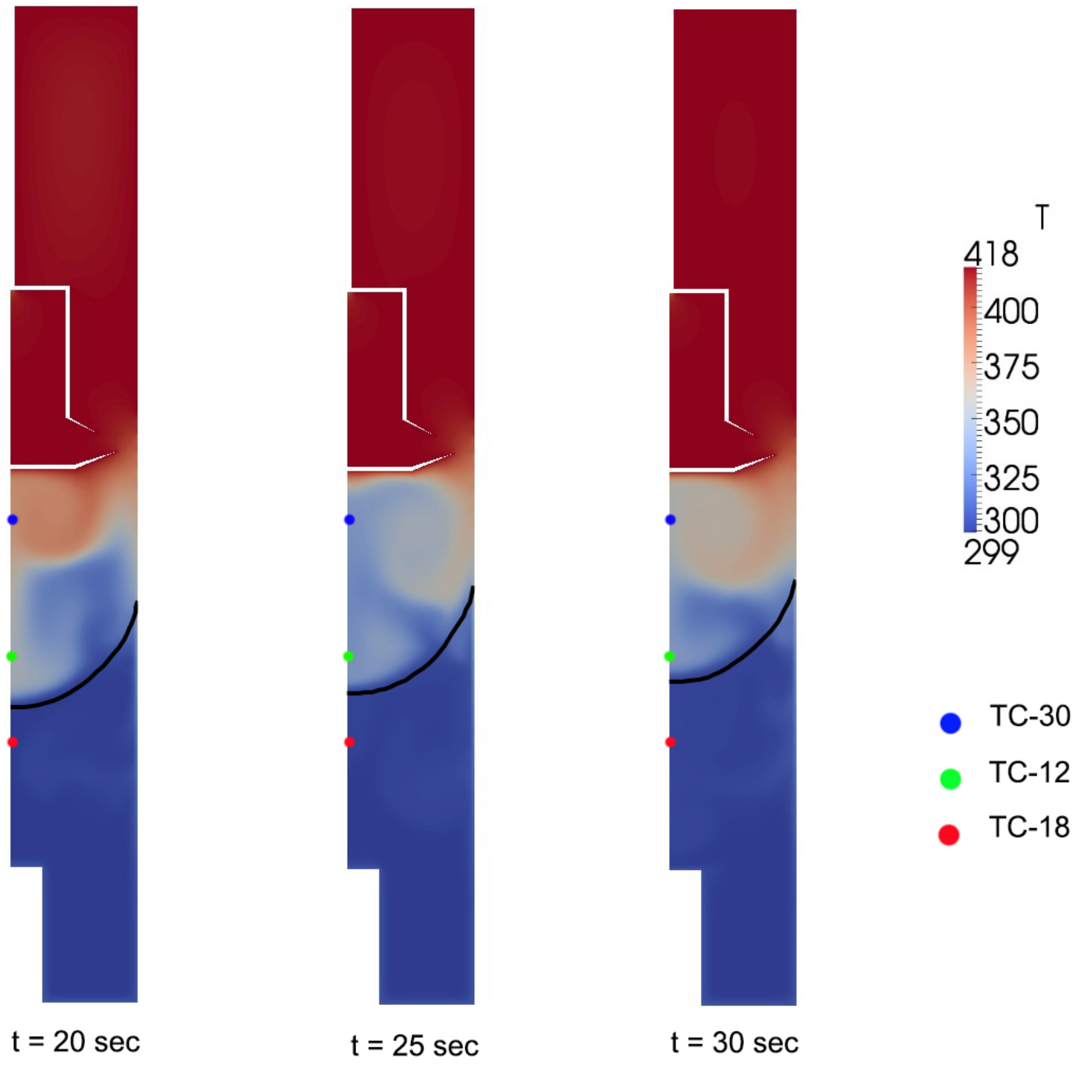


Figure 7.5: Temperature contours at 20, 25 and 30 seconds for case 1

As the inlet ports are closed during this phase, no mass enters or leaves test cell boundaries. From the temperature contours we observe that the vapor just above the liquid-vapor interface is quickly cooled down by the liquid. After 25 seconds we see an increase in the vapor temperature primarily near the wall because of the temperature boundary condition.

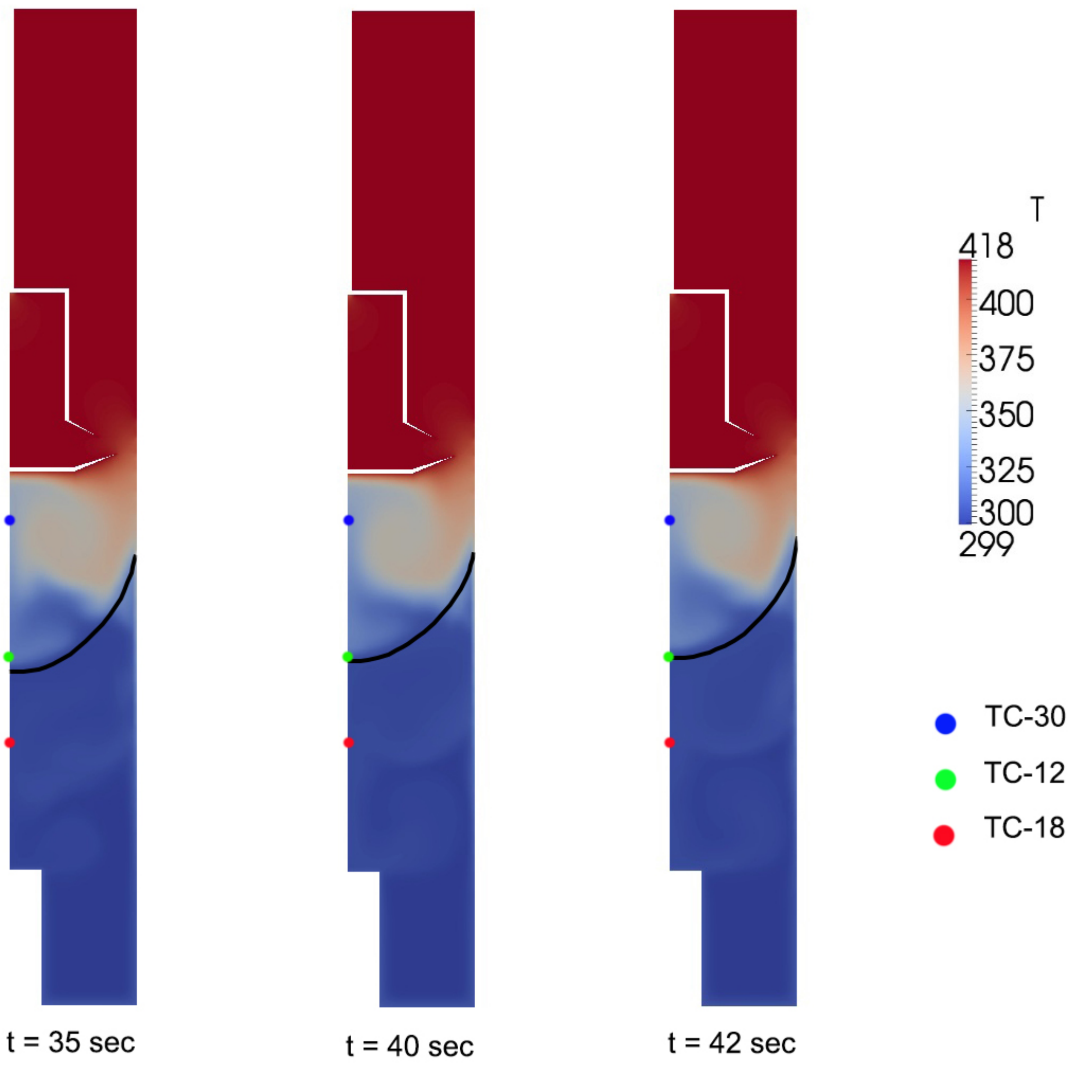
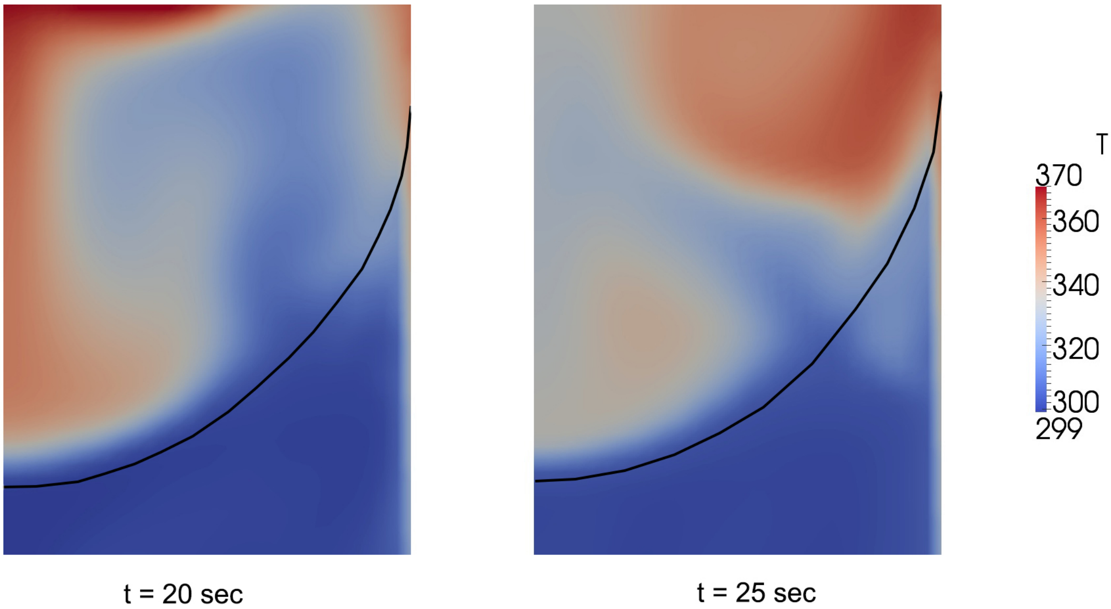


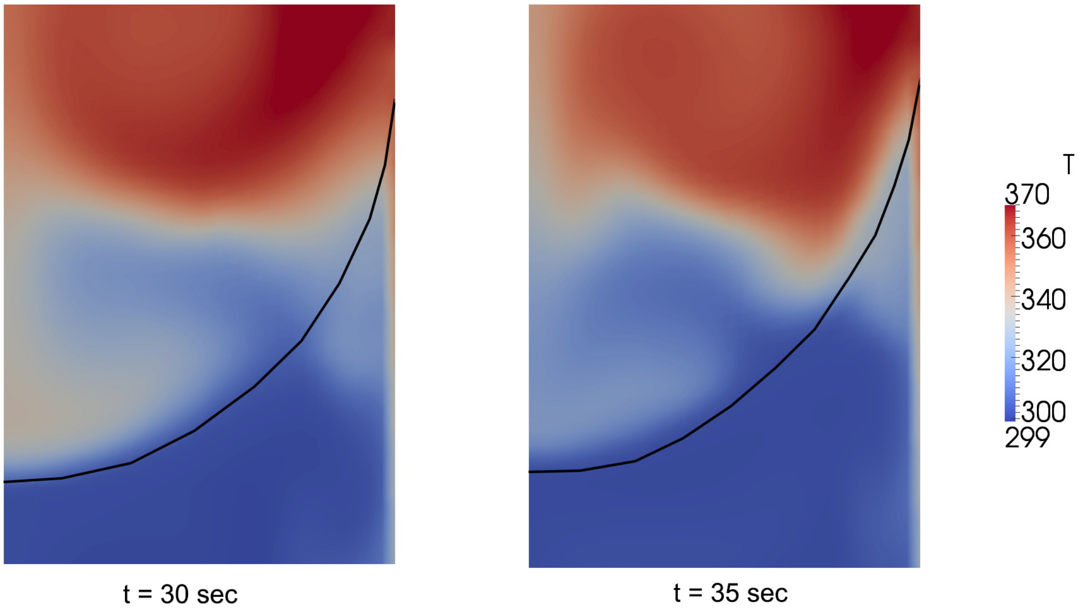
FIGURE 7.6: Temperature contours at 35, 40 and 42 seconds for case 1

The zoomed temperature contours at various times are shown in Figures 7.7, 7.8 and 7.9.


t = 20 sec

t = 25 sec

Figure 7.7: Temperature contours near the interface at 20 and 25 seconds for case 1


t = 30 sec

t = 35 sec

Figure 7.8: Temperature contours near the interface at 30 and 35 seconds for case 1

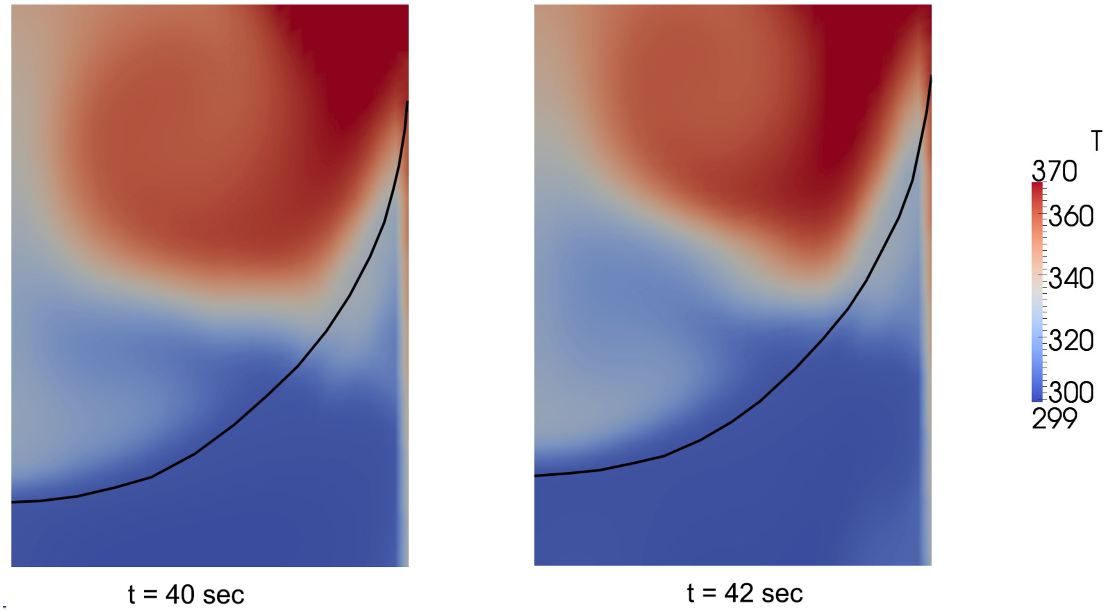


Figure 7.9: Temperature contours near the interface at 40 and 42 seconds for case 1

The temperatures during the second phase have been obtained at various locations. These locations correspond to the exact position of the thermocouples as in the SOURCE 2 experiment. The description and positions of the thermocouples are presented in the Table 7.2.

Thermocouple	Temperature Measurement	Position (mm)	
		x	y
TC-12	near interface	0	82
TC-18	of liquid	0	62
TC-30	of vapor	0	115

Table 7.2: Thermocouple locations in SOURCE 2 experiment

The temperature variation with time obtained from numerical simulations at the thermocouple locations mentioned in Table 7.2 is compared with the experimental results. Figure 7.10 shows how the vapor temperature varies in the vicinity of the liquid-vapor interface.

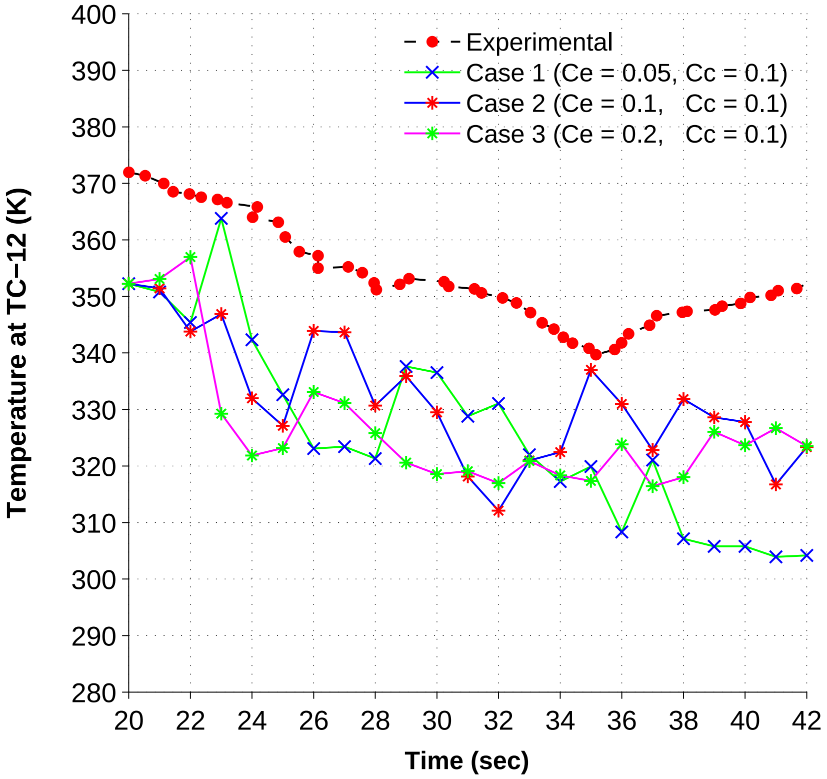


Figure 7.10: Temperature variation at TC-12

As can be seen from the plot, the temperatures obtained numerically seem to be quite lower than the experimental values. However, the general trend is in agreement as hot vapor gradually cools as it comes in contact with the cooler liquid. The fluctuations in vapor temperatures from the numerical simulations is very high as compared to the experiments. This could be attributed to the relatively high accommodation coefficients which cause rapid movement of vapor molecules.

If we have a closer look at the temperature from case 1, we see that the fluctuations die out after 38 seconds. However, this is not true for cases 2 and 3. Since the net condensation rate for case 1 is the highest, the liquid-vapor interface has risen to that level around 38 seconds where TC-12 is located. Therefore, the temperature is nearly equal to the bulk liquid temperature.

Next we are going to have a look at the temperature variation with time of the bulk liquid in Figure 7.11.

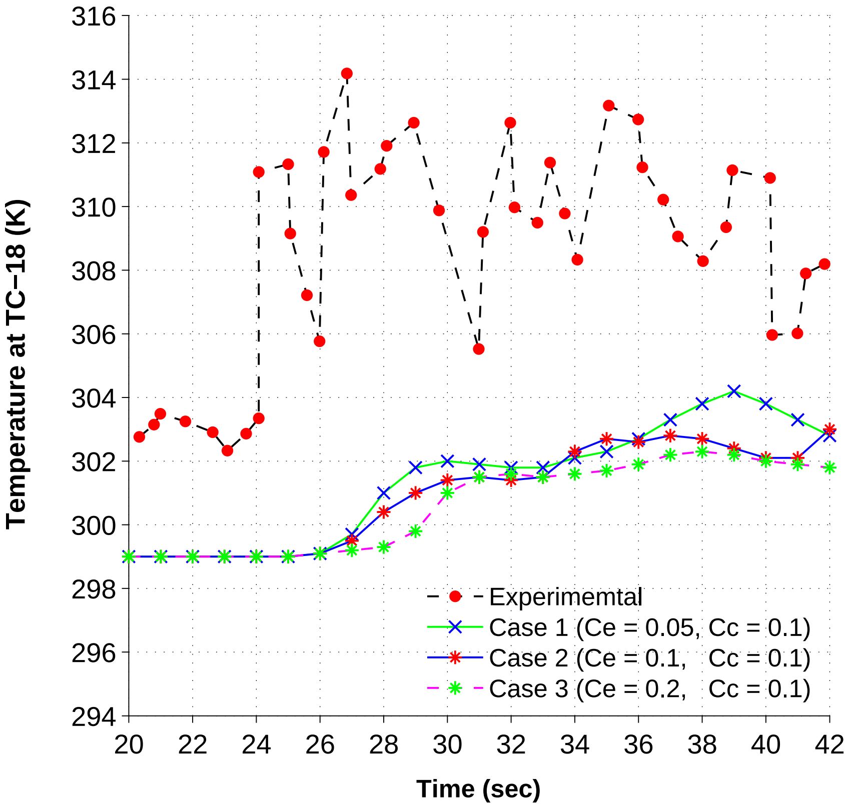


FIGURE 7.11: Temperature variation at TC-18

Here the situation is slightly different as the fluctuations in liquid temperatures obtained from experiments are extreme and difficult to explain. However, the numerical simulations yield much smoother results since the specific heat capacity of the liquid is much higher than the vapor (Appendix C). We can see from the plot that for all cases the temperature of the liquid starts to rise after 26 seconds. This rise is due to the heat gained from the hot vapor and the temperature boundary condition specified on the wall boundaries of tank r1.

We also observe from the trend that temperatures for case 1 are higher than case 2 which are in turn higher than case 3. This can be attributed to the increase in evaporation since the accommodation in case 1 is higher than in case 2 and case 3. Since there is more evaporation in case 3, there is more cooling of the surface and hence lower overall temperatures than the other two cases.

Lastly, we will have a look at the variation of bulk vapor temperature with time in Figure 7.12

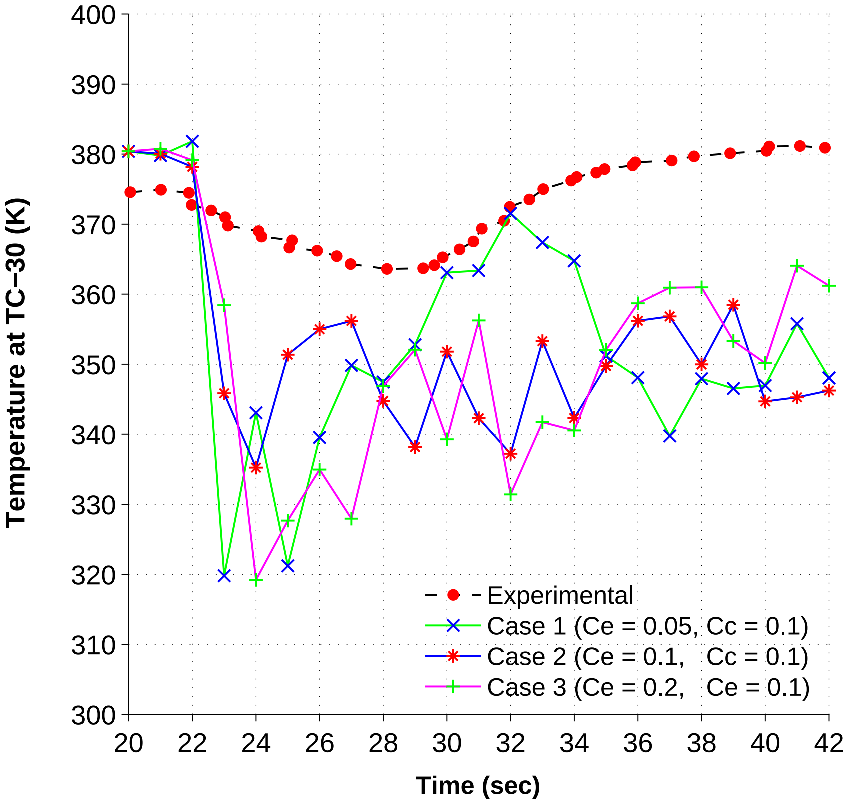


Figure 7.12: Temperature variation at TC-30

Here again we have a situation where the fluctuations in temperatures obtained numerically are very high. The reasons for this have already been discussed earlier. As can be seen from both the experimental and numerical results, the temperature starts to decrease initially as the vapor is cooled by the liquid. Afterwards, the temperature begins to rise again because of the temperature boundary condition specified on the walls of the test cell.

7.3 Liquid-Vapor Ratio

The liquid-vapor ratio is plotted below.

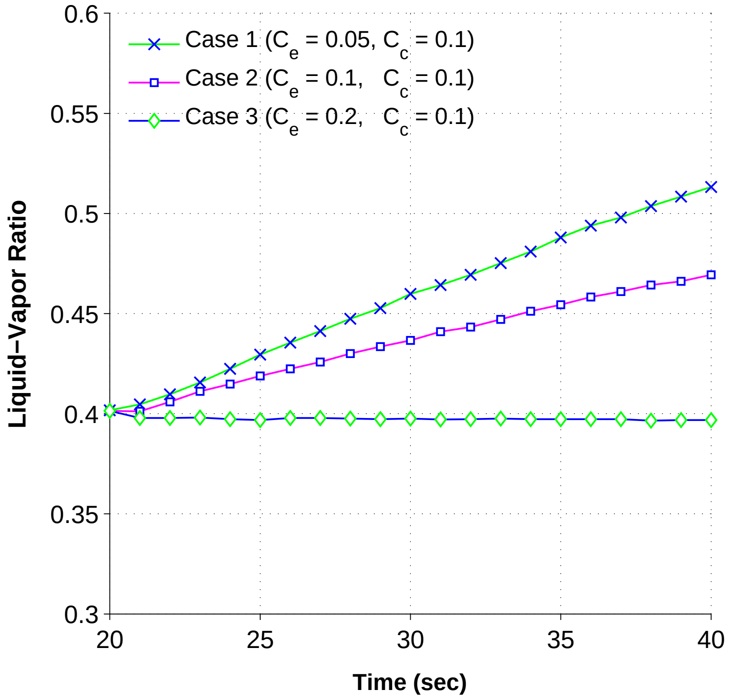


Figure 7.13: Liquid-vapor ratio for various accommodation coefficients

We can observe form the graph above that as the effective condensation increases from case 1 to case 3, we see that the liquid-vapor ratio tends to decrease.

Chapter 8

Summary and Outlook

In the present study, a phase change solver '*interEvapCondPhaseChangeFoam*' is developed in OpenFOAM which is capable of simulating interfacial evaporation and condensation phenomena. The solver is capable of simulating non isothermal flow between two immiscible fluids based on the Volume of Fluid method . The phase change process is based on the Kinetic Theory of Gases and is described by the Hertz-Knudsen equation. Fluid properties such as density, specific heat capacity, thermal conductivity, surface tension and kinematic viscosity are defined as a functions of temperature.

The solver is tested with a theoretical fluid in a Benchmark Evaporation Model subjected to microgravity conditions. The numerical simulations yielded satisfactory results with accommodation coefficient values for evaporation in the ranges $0.05 \leq C_e \leq 0.1$. It is found that the accommodation coefficient is dependent on the enthalpy of vaporization. As the enthalpy of vaporization is increased the accommodation coefficient needs to decreased so that the numerical and theoretical mass transfer rates are equal.

Finally an attempt is made to verify the phase change model based on a sounding rocket experiment (SOURCE 2). The first two phases of the experiment were numerically simulated and a comparison between the experimental and numerical results was obtained. The first phase comprised of pressurization of hot HFE-7000 vapor (418 K) from the top inlet and cold liquid (299 K) from the bottom inlet of SOURCE 2 test cell. At the beginning of the second phase both the inlet ports are closed so that there is no mass transfer across the test cell boundaries. Investigation of evaporation and condensation phenomena at the liquid-vapor interface has been carried out. Three test cases with different accommodation coefficients were simulated to study the effect on phase change. In general, the temperatures obtained are lower than the measured experimental values, however the trend is in agreement. Moreover, the simulation yielded results which are in compliance with the laws of Physics.

In future, several steps can be taken to model the phase change phenomena more accurately. Simulating cases with a wider range and combination of accommodation coefficients for evaporation and condensation would help understand better the phase change process. In addition, performing full three dimensional simulations would enable to avoid any discrepancies introduced due to symmetry boundary conditions. Since in this study HFE-7000 vapor was modelled as an ideal gas, utilizing its actual properties would allow us to observe the real gas behaviour. The actual geometry of SOURCE 2 test cell is complicated to some extent and therefore a simplified version was used in this thesis. By incorporating the exact complex geometry would enable the prediction of fluid behaviour more accurately. The study can also be extended by modelling the solid regions and solving the conjugate heat transfer in the whole domain.

Appendix A

1D Heat Transfer through an Interface

It has been shown by Patankar [22] that in the case of heat flux applied normal to the interface boundary, the expression for thermal conductivity in the interface region should be approximated by harmonic interpolation and not by linear interpolation. Here we consider a one dimensional case of heat transfer in two mediums denoted by 1 and 2. The properties in these mediums is denoted by the subscripts 1 and 2. The models is shown below:

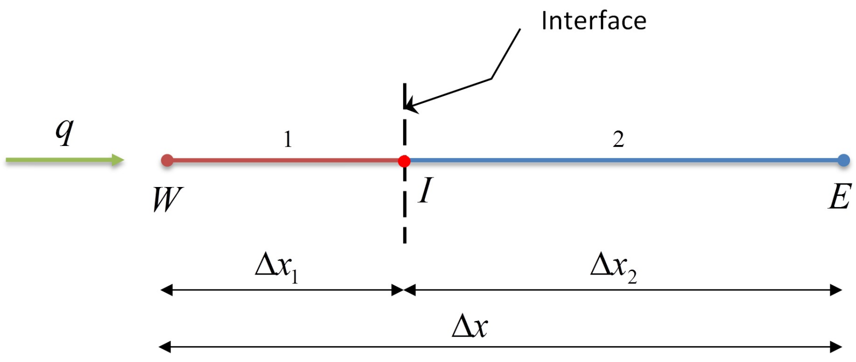


Figure A.1: One Dimensional Heat Transfer

The symbols W, I and E denote the West, Interface and East nodes respectively. From the Fourier's Law of Heat Conduction, the heat transferred throughout the medium can be found as:

$$q = \frac{k(T_W - T_E)}{\Delta x} \tag{A.1}$$

where k is the effective thermal conductivity of the two mediums.

We can also write the expressions for heat conduction in individual phases.

$$q_1 = \frac{k_1(T_W - T_I)}{\Delta x_1} \tag{A.2}$$

$$q_2 = \frac{k_2(T_I - T_E)}{\mathit{\Delta} x_2} \tag{A.3}$$

Multiply Eq (A.2) by $k_2/\mathit{\Delta} x_2$ and Eq (A.3) by $k_1/\mathit{\Delta} x_1$ we get

$$\frac{q_1 k_2}{\mathit{\Delta} x_2} = \frac{k_1 k_2(T_W - T_I)}{\mathit{\Delta} x_1 \mathit{\Delta} x_2} \tag{A.4}$$

$$\frac{q_2 k_1}{\mathit{\Delta} x_1} = \frac{k_1 k_2(T_I - T_E)}{\mathit{\Delta} x_1 \mathit{\Delta} x_2} \tag{A.5}$$

Adding the above equations results in:

$$\frac{q_1 k_2}{\mathit{\Delta} x_2} + \frac{q_2 k_1}{\mathit{\Delta} x_1} = \frac{k_1 k_2(T_W - T_E)}{\mathit{\Delta} x_1 \mathit{\Delta} x_2} \tag{A.6}$$

We also know by conservation of energy that $q_1 = q_2 = q$. So we can write:

$$q\left(\frac{k_2}{\mathit{\Delta} x_2} + \frac{k_1}{\mathit{\Delta} x_1}\right) = \frac{k_1 k_2(T_W - T_E)}{\mathit{\Delta} x_1 \mathit{\Delta} x_2} \tag{A.7}$$

$$q = \frac{k_1 k_2(T_W - T_E)}{k_1 \mathit{\Delta} x_2 + k_2 \mathit{\Delta} x_1} \tag{A.8}$$

So now comparing with Eq (A.1) we have:

$$k = \frac{k_1 k_2}{k_1 \frac{\mathit{\Delta} x_2}{\mathit{\Delta} x} + k_2 \frac{\mathit{\Delta} x_1}{\mathit{\Delta} x}} \tag{A.9}$$

Since we know that:

$$\mathit{\Delta} x = \mathit{\Delta} x_1 + \mathit{\Delta} x_2 \tag{A.10}$$

We can modify Eq (A.9) as:

$$k = \frac{k_1 k_2}{k_1\left(1 - \frac{\mathit{\Delta} x_1}{\mathit{\Delta} x}\right) + k_2 \frac{\mathit{\Delta} x_1}{\mathit{\Delta} x}} \tag{A.11}$$

By considering the length fraction, $\gamma_1 = \frac{\mathit{\Delta} x_1}{\mathit{\Delta} x}$, we can rewrite the expression as:

$$k = \frac{k_1 k_2}{k_1\left(1 - \gamma_1\right) + k_2 \gamma_1} \tag{A.12}$$

An equivalent expression can be assumed for a three dimensional case with volume phase fraction for phase 1.

$$k = \frac{k_1 k_2}{k_1\left(1 - \alpha_1\right) + k_2 \alpha_1} \tag{A.13}$$

Appendix B

Tutorials

The steps followed frequently during the thesis include creation of geometry in GMSH, modyfying OpenFOAM solvers and libraries. A guideline is provided in this Appendix for the procedure. The text in **bold** indicates that these are Linux commands.

B.1 Geometry Creation in Gmsh

In this section we are going to discuss the creation of geometry in Gmsh and the steps to convert it to OpenFOAM format.

STEP 1: OPEN GMSH
Go to the directory of the case file and run GMSH by typing:
gmsh

STEP 2: CREATE GEOMETRY
Geometry can be created in GMSH through the Graphical User Interface (GUI) or with the help of a text file with a '.geo' extension.

STEP 3: CREATE MESH
Now create a 3D mesh and save the file. This is necessary as OpenFOAM only recognizes three dimensional meshes.

STEP 4: CONVERT GMSH FORMAT TO OPENFOAM FORMAT
We assume that the our GMSH mesh file is named 'testcase.msh'. In order to convert the mesh created in GMSH to OpenFOAM format the following command must be typed:
gmshToFoam testcase.msh

STEP 5: CHANGE THE 'BOUNDARY' FILE

Go to the following directory and open the 'boundary' file.

```
cd constant/polyMesh
```

Change the 'patch type' field for all patches which should be consistent with the boundary conditions.

STEP 6: CHANGE THE 'ALPHA1' FILE

Go two levels up and then change directory to:

```
cd 0
```

Delete the file 'alpha1' if already present

```
rm alpha1
```

Then copy the original file for alpha1 field ‘alpha1.org‘ to 'alpha1'.

```
cp alpha1.org alpha1
```

The case is now ready to be solved.

B.2 Modifying OpenFOAM Solvers

This section will focus on how to make changes to the existing OpenFOAM solvers in order to suit our requirements.

STEP 1: CREATE DIRECTORY TO COPY EXISTING SOLVERS
This is necessary as to avoid tampering the original OpenFOAM solvers. In this step we are copying the solver files to the user’s home directory ($WM_PROJECT_USER_DIR). If the user intends to copy the files to another directory, the appropriate path must be used. This step should be skipped if the directory where the files are to be copied already exists.

```
mkdir -p $WM_PROJECT_USER_DIR/applications/solvers
```

STEP 2: COPY OPENFOAM SOLVERS
Copy the existing OPENFOAM solvers to the intended directory.

```
cp -r $FOAM_SOLVERS $WM_PROJECT_USER_DIR/applications/solvers
```

STEP 3: CHANGE THE NAME OF THE SOLVER
Go to the directory of the type of solver you want to modify. We are assuming that we want to modify ’compressibleInterFoam’ which is a multiphase solver.

```
cd $WM_PROJECT_USER_DIR/applications/solvers/multiphase
```

Change the name of the solver to something which describes what additional features this new solver will include. For instance, if we are adding the energy equation to solve for the temperature field in ’compressibleInterFoam’, we can rename it to ’compressibleInterTempFoam’. This is necessary for ease of the user.

```
mv compressibleInterFoam compressibleInterTempFoam
```

STEP 4: CHANGE THE NAME OF THE SOURCE FILE
Go to the directory of the solver.

```
cd compressibleInterTempFoam
```

The source file should be renamed according to the name of the solver.

```
mv compressibleInterFoam.C compressibleInterTempFoam.C
```

STEP 5: CHECK THE SOURCE FILES

Open the source file and check if there are any header files included by a relative path. If such is the case, you should correct the new path.

STEP 6: EDIT THE SOURCE FILE

Edit the source file to incorporate the modifications intended in the new solver.

STEP 7: CLEAN THE FILES

This is necessary to remove dependencies of the files coming from the original solver.

wclean

STEP 8: EDIT THE MAKE/FILES

Go to the make directory.

cd Make

Edit the 'files' to incorporate the new name and path of the solver. For compressibleInterFoam the 'files' originally was:

```
compressibleInterFoam.C
EXE = $(FOAM_APPBIN)/compressibleInterFoam
```

Change it to:

```
compressibleInterTempFoam.C
EXE = $(FOAM_USER_APPBIN)/compressibleInterTempFoam
```

STEP 9: EDIT THE OPTIONS/FILES

Edit the 'options' file if you have made any changes to the OpenFOAM libraries.

STEP 10: COMPILE THE NEW SOLVER

Go to the directory a level above and type

wmake

The modified solver is now compiled and ready to solve a case.

B.3 Modifying OpenFOAM Libraries

This section will focus on how to make changes to the existing OpenFOAM libraries in order to suit our requirements. It is assumed that the user has already performed STEP 1 of ’Modifying OpenFOAM Solvers’

STEP 1: CREATE DIRECTORY TO COPY EXISTING SOLVERS
Copy the existing OPENFOAM libraries to the intended directory.
cp -r $FOAM_SRC $WM_PROJECT_USER_DIR/src

STEP 2: EDIT THE LIBRARY
Go to the directory of the library you want to modify. For instance if we want to modify ’interfaceProperties’, then type
cd $WM_PROJECT_USER_DIR/applications/src/transportModels/ interfaceProperties

Open the relevant header and source files and modify them.

STEP 3: EDIT THE MAKE/FILES
Go to the make directory.
cd Make

Edit the ’files’ to incorporate the new name and path of the solver. For ’interfaceProperties’ the ’files’ originally was:
interfaceProperties.C
interfaceCompression/interfaceCompression.C
LIB = $(FOAM_LIBBBIN)/libInterfaceProperties

Change it to:
interfaceProperties.C
interfaceCompression/interfaceCompression.C
LIB = $(FOAM_USER_LIBBBIN)/libmyInterfaceProperties

STEP 4: EDIT THE OPTIONS/FILES

Edit the 'options' file if any changes need to be made.

STEP 5: CLEAN AND COMPILE THE MODIFIED LIBRARY

Go to the directory a level above and type

wclean libso && wmake libso

Appendix C

HFE 7000 Properties

The properties of the working fluid in the SOURCE 2 experiment is presented in this Appendix. The variation of these properties with respect to temperature and pressure has been utilized in the simulations.

General Properites

Molar mass = 0.2 kg/mol
The critical properties of HFE 7000 [23] are listed below:

$$\begin{aligned} T_c &= 437.6\ K \\ p_c &= 2483.3\ kPa \\ \rho_c &= 553\ kg/m^3 \end{aligned} \tag{C.1}$$

Density

The density dependence on temperature for HFE 7000 liquid is given by the relation [23]:

$$\rho_l = 2258 - 2.88\,T \tag{C.2}$$

For the vapor phase the Ideal Gas Law is used.

$$\rho_v = \frac{pM}{R_uT} = \frac{p}{41.57\ T} \tag{C.3}$$

Specific Heat Capacity

The specific heat capacity dependence on temperature for HFE 7000 is given by the relations:

$$c_{p_l} = 381.89 + 3.08\,T \tag{C.4}$$

$$c_{p_v} = (3.68 * 10^{-6})\,T^3 - (5.44 * 10^{-3})\,T^2 + 4.14\,T - 0.166 \tag{C.5}$$

The relation for vapor phase holds true for 260K < T < 420K.

Thermal Conductivity

The thermal conductivity dependence on temperature for HFE 7000 liquid is given by the relation:

$$k_l = 0.133 - 0.000196\,T \tag{C.6}$$

For HFE 7000 vapor the thermal conductivity can be estimated from the modified Eucken expression for polyatomic gases [24].

$$\begin{aligned} k_v &= C_{p_v}\mu_v + 1.25\left(\frac{R_u\mu_v}{M}\right) \\ k_v &= C_{p_v}\rho_v\nu_v + 52\rho_v\nu_v \end{aligned} \tag{C.7}$$

Surface Tension

The surface tension dependence on temperature for HFE 7000 is given by the relation:

$$\sigma = 0.0428\left(1 - \frac{T}{T_c}\right)^{1.016} \tag{C.8}$$

Kinematic Viscosity

The kinematic viscosity dependence on temperature for HFE 7000 liquid is given by the relation provided by the manufacturer [23]:

$$\nu_l = 10^{-6}\left(z - e^{-0.7487-3.295z+0.6199z^2-0.3193z^3}\right) \tag{C.9}$$

where $z = 10^{10^{10.151-4.6006\log(T)}} - 0.7$

The estimate for the vapor viscosity (in Poise) can be found using the corresponding states method described in Lucas.

$$\nu_v = \frac{10^{-6}}{\rho_v}\zeta^{-1}\left(0.807T_r^{0.618} - 0.357e^{-0.449T_r} + 0.340e^{-4.058T_r} + 0.018\right) \tag{C.10}$$

where $\zeta = 0.176\left(\frac{T_c}{M^3P_c^4}\right)^{1/6}$

In the above expressions molar mass is in g/mol, temperature is in Kelvin and pressure is in bars. In SI units the expression for ζ becomes:

$$\begin{aligned}\zeta &= 0.176\left(\frac{T_c}{(1000\ M)^3\left(\frac{P_c}{100000}\right)^4}\right)^{1/6}\\ \zeta &= 4.032\times10^{-3}\end{aligned} \tag{C.11}$$

Now we can simplify the expression for vapor kinematic viscosity in SI units.

$$\nu_v = \frac{1}{\rho_v}\left(0.0002T_r^{0.618} - 0.000088e^{-0.449T_r} + 0.000084e^{-4.058T_r} + 0.00000446\right) \tag{C.12}$$

Saturation Pressure

The variation of saturated vapor pressure is given by the relation [23]:

$$p_{sat} = e^{\left(\frac{-3548.6}{T}+22.978\right)} \tag{C.13}$$

This expression is valid for the temperature range of $243K < T < T_c$.

Appendix D

Codes

D.1 Source 2 Geometry

The code for creating the SOURCE 2 geometry in Gmsh software is presented below.

```
// PARAMETER FOR CONTROLLING MESH SIZE
   lc = 0.01;

// SECTION 1

   // Defining Labels
      section = 1;
      no_p = 0;
      no_l = 0;

   // Defining Points and Lines
      Point(no_p + 1) = {0.0075, 0, 0, lc};
      Point(no_p + 2) = {0.03, 0, 0, lc};
      Point(no_p + 3) = {0.03, 0.004, 0, lc};
      Point(no_p + 4) = {0.03, 0.022, 0, lc};
      Point(no_p + 5) = {0.03, 0.027, 0, lc};
      Point(no_p + 6) = {0.03, 0.032, 0, lc};
      Point(no_p + 7) = {0.0075, 0.032, 0, lc};

      Line(no_l + 1) = {1,2};
      Line(no_l + 2) = {2,3};
      Line(no_l + 3) = {3,4};
      Line(no_l + 4) = {4,5};
      Line(no_l + 5) = {5,6};
      Line(no_l + 6) = {6,7};
      Line(no_l + 7) = {7,1};



   // Defining Line Loop and Surface
      Line Loop(section) = {1,2,3,4,5,6,7} ;
      Plane Surface(section) = {section};

   // Defining Number of Nodes on Lines
      Transfinite Line{1,6} = 23;
      Transfinite Line{2} = 5;
      Transfinite Line{3} = 19;
      Transfinite Line{4} = 6;
      Transfinite Line{5} = 6;
      Transfinite Line{7} = 33;

   // Defining Corners of the Surface to create a Structured Mesh
      Transfinite Surface{section} = {1,2,6,7};
      Recombine Surface{section};

// SECTION 2

   // Defining Labels
      section = 2;
      no_p = 7;
      no_l = 7;

   // Defining Points and Lines
      Point(no_p + 1) = {0.03, 0.036, 0, lc};
      Point(no_p + 2) = {0.03, 0.041, 0, lc};
      Point(no_p + 3) = {0.03, 0.046, 0, lc};
      Point(no_p + 4) = {0.03, 0.051, 0, lc};
      Point(no_p + 5) = {0.03, 0.056, 0, lc};
      Point(no_p + 6) = {0.03, 0.061, 0, lc};
      Point(no_p + 7) = {0.03, 0.066, 0, lc};
      Point(no_p + 8) = {0.03, 0.071, 0, lc};
      Point(no_p + 9) = {0.03, 0.076, 0, lc};
      Point(no_p + 10) = {0.03, 0.081, 0, lc};
      Point(no_p + 11) = {0.03, 0.086, 0, lc};
      Point(no_p + 12) = {0.03, 0.091, 0, lc};
      Point(no_p + 13) = {0.03, 0.096, 0, lc};
      Point(no_p + 14) = {0.03, 0.101, 0, lc};
      Point(no_p + 15) = {0.03, 0.106, 0, lc};
      Point(no_p + 16) = {0.03, 0.111, 0, lc};
      Point(no_p + 17) = {0.03, 0.116, 0, lc};
      Point(no_p + 18) = {0.03, 0.121, 0, lc};
      Point(no_p + 19) = {0.03, 0.126, 0, lc};
```

```
       Point(no_p + 20) = {0.03, 0.13, 0, lc};
       Point(no_p + 21) = {0.025, 0.13, 0, lc};
       Point(no_p + 22) = {0.0155, 0.126, 0, lc};
       Point(no_p + 23) = {0.0, 0.126, 0, lc};
       Point(no_p + 24) = {0.0, 0.032, 0, lc};
       Line(no_l + 1) = {6,8};
       Line(no_l + 2) = {8,9};
       Line(no_l + 3) = {9,10};
       Line(no_l + 4) = {10,11};
       Line(no_l + 5) = {11,12};
       Line(no_l + 6) = {12,13};
       Line(no_l + 7) = {13,14};
       Line(no_l + 8) = {14,15};
       Line(no_l + 9) = {15,16};
       Line(no_l + 10) = {16,17};
       Line(no_l + 11) = {17,18};
       Line(no_l + 12) = {18,19};
       Line(no_l + 13) = {19,20};
       Line(no_l + 14) = {20,21};
       Line(no_l + 15) = {21,22};
       Line(no_l + 16) = {22,23};
       Line(no_l + 17) = {23,24};
       Line(no_l + 18) = {24,25};
       Line(no_l + 19) = {25,26};
       Line(no_l + 20) = {26,27};
       Line(no_l + 21) = {27,28};
       Line(no_l + 22) = {28,29};
       Line(no_l + 23) = {29,30};
       Line(no_l + 24) = {30,31};
       Line(no_l + 25) = {31,7};

    // Defining Line Loop and Surface
       Line Loop(section) = {-6,8,9,10,11,12,13,14,15,16,17,18,19,20
       ,21,22,23,24,25,26,27,28,29,30,31,32};
       Plane Surface(section) = {section};

    // Defining Number of Nodes on Lines
       Transfinite Line{no_l + 1} = 5;
       Transfinite Line{no_l + 2} = 6;
       Transfinite Line{no_l + 3} = 6;
       Transfinite Line{no_l + 4} = 6;
       Transfinite Line{no_l + 5} = 6;
       Transfinite Line{no_l + 6} = 6;
```

```
      Transfinite Line{no_l + 7} = 6;
      Transfinite Line{no_l + 8} = 6;
      Transfinite Line{no_l + 9} = 6;
      Transfinite Line{no_l + 10} = 6;
      Transfinite Line{no_l + 11} = 6;
      Transfinite Line{no_l + 12} = 6;
      Transfinite Line{no_l + 13} = 6;
      Transfinite Line{no_l + 14} = 6;
      Transfinite Line{no_l + 15} = 6;
      Transfinite Line{no_l + 16} = 6;
      Transfinite Line{no_l + 17} = 6;
      Transfinite Line{no_l + 18} = 6;
      Transfinite Line{no_l + 19} = 6;
      Transfinite Line{no_l + 20} = 5;
      Transfinite Line{no_l + 21} = 6;
      Transfinite Line{no_l + 22} = 11;
      Transfinite Line{no_l + 23} = 15;
      Transfinite Line{no_l + 24} = 99;
      Transfinite Line{no_l + 25} = 8;

   // Defining Corners of the Surface to create a Structured Mesh
      Transfinite Surface{section} = {6,27,30,31};
      Recombine Surface{section};

// SECTION 3

   // Defining Labels
      section = 3;
      no_p = 31;
      no_l = 32;

   // Defining Points and Lines
      Point(no_p + 1) = {0.03, 0.134, 0, lc};
      Point(no_p + 2) = {0.021, 0.134, 0, lc};
      Point(no_p + 3) = {0.0155, 0.127, 0, lc};
      Line(no_l + 1) = {27,32};
      Line(no_l + 2) = {32,33};
      Line(no_l + 3) = {33,34};
      Line(no_l + 4) = {34,28};

   // Defining Line Loop and Surface
      Line Loop(section) = {-28,33,34,35,36} ;
      Plane Surface(section) = {section};

   // Defining Number of Nodes on Lines
      Transfinite Line{no_l + 1} =  10;
      Transfinite Line{no_l + 2} = 13;
      Transfinite Line{no_l + 3} = 10;
      Transfinite Line{no_l + 4} = 8;

   // Defining Corners of the Surface to create a Structured Mesh
      Transfinite Surface{section} = {27,32,33,34};
      Recombine Surface{section};

// SECTION 4

   // Defining Labels
      section = 4;
      no_p = 35;
      no_l = 36;

   // Defining Points and Lines
      Point(no_p + 1) = {0, 0.134, 0, lc};
      Point(no_p + 2) = {0, 0.127, 0, lc};
      Line(no_l + 1) = {33,36};
      Line(no_l + 2) = {36,37};
      Line(no_l + 3) = {37,34};

   // Defining Line Loop and Surface
      Line Loop(section) = {-35,37,38,39} ;
      Plane Surface(section) = {section};

   // Defining Number of Nodes on Lines
      Transfinite Line{no_l + 1} = 18;
      Transfinite Line{no_l + 2} = 10;
      Transfinite Line{no_l + 3} = 18;

   // Defining Corners of the Surface to create a Structured Mesh
      Transfinite Surface{section} = {37,34,33,36};
      Recombine Surface{section};

// SECTION 5

   // Defining Labels
      section = 5;
      no_p = 37;
      no_l = 39;

   // Defining Points and Lines
      Point(no_p + 1) = {0.03, 0.138, 0, lc};
      Point(no_p + 2) = {0.03, 0.169, 0, lc};
      Point(no_p + 3) = {0.014, 0.169, 0, lc};
      Point(no_p + 4) = {0.014, 0.138, 0, lc};
      Line(no_l + 1) = {32,38};
      Line(no_l + 2) = {38,39};
      Line(no_l + 3) = {39,40};
      Line(no_l + 4) = {40,41};
      Line(no_l + 5) = {41,33};

   // Defining Line Loop and Surface
      Line Loop(section) = {-34,40,41,42,43,44} ;
      Plane Surface(section) = {section};

   // Defining Number of Nodes on Lines
      Transfinite Line{no_l + 1} = 7;
      Transfinite Line{no_l + 2} = 28;
      Transfinite Line{no_l + 3} = 21;
      Transfinite Line{no_l + 4} = 34;
      Transfinite Line{no_l + 5} = 9;

   // Defining Corners of the Surface to create a Structured Mesh
      Transfinite Surface{section} = {41,32,39,40};
      Recombine Surface{section};

// SECTION 6

   // Defining Labels
      section = 6;
      no_p = 41;
      no_l = 44;

   // Defining Points and Lines
      Point(no_p + 1) = {0.013, 0.138, 0, lc};
      Point(no_p + 2) = {0, 0.138, 0, lc};
      Line(no_l + 1) = {33,42};
      Line(no_l + 2) = {42,43};
      Line(no_l + 3) = {43,36};


```

```
    // Defining Line Loop and Surface
      Line Loop(section) = {-37,45,46,47} ;
      Plane Surface(section) = {section};

    // Defining Number of Nodes on Lines
      Transfinite Line{no_l + 1} = 7;
      Transfinite Line{no_l + 2} = 18;
      Transfinite Line{no_l + 3} = 7;

    // Defining Corners of the Surface to create a Structured Mesh
      Transfinite Surface{section} = {36,33,42,43};
      Recombine Surface{section};

// SECTION 7

    // Defining Labels
      section = 7;
      no_p = 43;
      no_l = 47;

    // Defining Points and Lines
      Point(no_p + 1) = {0.013, 0.168, 0, lc};
      Point(no_p + 2) = {0.001, 0.168, 0, lc};
      Point(no_p + 3) = {0, 0.168, 0, lc};
      Line(no_l + 1) = {42,44};
      Line(no_l + 2) = {44,45};
      Line(no_l + 3) = {45,46};
      Line(no_l + 4) = {46,43};

    // Defining Line Loop and Surface
      Line Loop(section) = {-46,48,49,50,51} ;
      Plane Surface(section) = {section};

    // Defining Number of Nodes on Lines
      Transfinite Line{no_l + 1} = 35;
      Transfinite Line{no_l + 2} = 16;
      Transfinite Line{no_l + 3} = 3;
      Transfinite Line{no_l + 4} = 35;

    // Defining Corners of the Surface to create a Structured Mesh
      Transfinite Surface{section} = {43,42,44,46};
      Recombine Surface{section};

```

```
// SECTION 8

   // Defining Labels
      section = 8;
      no_p = 46;
      no_l = 51;

   // Defining Points and Lines
      Point(no_p + 1) = {0.03, 0.235, 0, lc};
      Point(no_p + 2) = {0.001, 0.235, 0, lc};
      Point(no_p + 3) = {0.001, 0.169, 0, lc};
      Line(no_l + 1) = {39,47};
      Line(no_l + 2) = {47,48};
      Line(no_l + 3) = {48,49};
      Line(no_l + 4) = {49,40};

   // Defining Line Loop and Surface
      Line Loop(section) = {-42,52,53,54,55} ;
      Plane Surface(section) = {section};

   // Defining Number of Nodes on Lines
      Transfinite Line{no_l + 1} = 60;
      Transfinite Line{no_l + 2} = 35;
      Transfinite Line{no_l + 3} = 60;
      Transfinite Line{no_l + 4} = 15;

   // Defining Corners of the Surface to create a Structured Mesh
      Transfinite Surface{section} = {49,39,47,48};
      Recombine Surface{section};


// 3D EXTRUSION
      Extrude{0,0,0.01}{
       Surface{1};Surface{2};Surface{3};Surface{4};Surface{5};Surface{6};
   Surface{7};Surface{8};
      Layers{1};Recombine;
      }

// DEFINING SURFACES FOR BOUNDARY CONDITIONS

   // Inlet Boundaries
      Physical Surface("Bottom_Inlet") = {71};
      Physical Surface("Top_Inlet") = {349};
   // Axis of Symmetry
      Physical Surface("R1_Axis") = {219,268,326,353};

   // Tank r1 Walls
      Physical Surface("R1_Right_Wall_1") = {79};
      Physical Surface("R1_Right_Wall_2") = {83};
      Physical Surface("R1_Right_Wall_3") = {127};
      Physical Surface("R1_Right_Wall_4") = {131};
      Physical Surface("R1_Right_Wall_5") = {135};
      Physical Surface("R1_Right_Wall_6") = {139};
      Physical Surface("R1_Right_Wall_7") = {143};
      Physical Surface("R1_Right_Wall_8") = {147};
      Physical Surface("R1_Right_Wall_9") = {151};
      Physical Surface("R1_Right_Wall_10") = {155};
      Physical Surface("R1_Right_Wall_11") = {159};
      Physical Surface("R1_Right_Wall_12") = {163};
      Physical Surface("R1_Right_Wall_13") = {167};
      Physical Surface("R1_Right_Wall_14") = {171};
      Physical Surface("R1_Right_Wall_15") = {175};
      Physical Surface("R1_Right_Wall_16") = {179};
      Physical Surface("R1_Right_Wall_17") = {183};
      Physical Surface("R1_Right_Wall_18") = {187};
      Physical Surface("R1_Right_Wall_19") = {191};
      Physical Surface("R1_Right_Wall_20") = {195};
      Physical Surface("R1_Right_Wall_21") = {199};
      Physical Surface("R1_Right_Wall_22") = {203};
      Physical Surface("R1_Right_Wall_23") = {238};
      Physical Surface("R1_Right_Wall_24") = {288};

   // GPPS Walls
      Physical Surface("GPPS_Vertical_Wall") = {341,300};
      Physical Surface("GPPS_Slanted_Bottom_Wall") = {250,304};
      Physical Surface("GPPS_Slanted_Top_Wall") = {318,211};
      Physical Surface("GPPS_Horizontal_Wall") = {272,215,345};

   // Bottom Compartment Walls
      Physical Surface("Bottom_Component_Bottom_Wall") = {67,223};
      Physical Surface("Bottom_Component_Left_Wall") = {91};
      Physical Surface("Bottom_Component_Right_Wall") = {75};

   // Top Compartment Walls
      Physical Surface("Top_Component_Right_Wall") = {292,368};
      Physical Surface("Top_Component_Left_Wall") = {376};
      Physical Surface("Top_Component_Top_Wall") = {372};
      Physical Surface("Top_Component_Bottom_Wall") = {380};

// DEFINING VOLUME
      Physical Volume("volume") = {1,2,3,4,5,6,7,8};
```

D.2 interEvapCondPhaseChangeFoam Solver

The source code of the main solver 'interEvapCondPhaseChangeFoam.C' with self explanatory comments is shown here.

```
/*---------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------



License
   This file is part of OpenFOAM. OpenFOAM is free software: you can
   redistribute it and/or modify it under the terms of the GNU General
   Public License as published by the Free Software Foundation, either
   version 3 of the License, or (at your option) any later version.
   OpenFOAM is distributed in the hope that it will be useful, but
   WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
   General Public License for more details. You should have received
   a copy of the GNU General Public License along with OpenFOAM. If not,
   see <http://www.gnu.org/licenses/>.

Application
   interEvapCondPhaseChangeFoam

Description
   A solver for 2 compressible, non-isothermal immiscible fluids with
   capable of simulating interfacial evaporation and condensation. It
   utilizes a VOF (volume of fluid) phase-fraction based interface
   capturing approach. The momentum and other fluid properties are of
   the "mixture" and a single momentum equation is solved. The fluid
   properties such as density, viscosity, specific heat, thermal
   conductivity and surface tension are temperature dependent.
   The phase change model is based on the Hertz-Knudsen equation.


\*---------------------------------------------------------------------*/
```

```
// INCLUSION OF RELEVANT HEADER FILES
#include "fvCFD.H"
#include "MULES.H"
#include "subCycle.H"
#include "interfaceProperties.H"
#include "phaseChangeTwoPhaseMixture.H"
#include "turbulenceModel.H"
#include "pimpleControl.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

// MAIN PROGRAM BEGINS
int main(int argc, char *argv[])
{
   // INCLUSION OF THE RELEVANT HEADER FILES INSIDE THE MAIN PROGRAM
   #include "setRootCase.H"
   #include "createTime.H"
   #include "createMesh.H"
   #include "readGravitationalAcceleration.H"
   #include "initContinuityErrs.H"
   #include "createFields.H"
   #include "readTimeControls.H"
   #include "correctPhi.H"
   #include "CourantNo.H"
   #include "setInitialDeltaT.H"
   #include "calcPSatField.H"

   pimpleControl pimple(mesh);

   // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

   Info<< "\nStarting time loop\n" << endl;

   // TIME LOOP
   while (runTime.run())
   {
      #include "readTimeControls.H"
      #include "CourantNo.H"
      #include "setDeltaT.H"

      runTime++;
      turbulence->correct();

      // PRESSURE-VELOCITY PIMPLE CORRECTOR LOOP
      while (pimple.loop())
      {
         #include "alphaEqnSubCycle.H"
         if (pimple.corr() == 1)
         {
            interface.correct();
         }

         // MASS CONSERVATION ON THE WHOLE SYSTEM
         solve(fvm::ddt(rho) + fvc::div(rhoPhi));

         // SOLVING FOR THE VELOCITY FIELD
         #include "UEqn.H"

         // SOLVING FOR TEMPERATURE FIELD
         #include "TEqn.H"

         // PRESSURE CORRECTOR LOOP
         while (pimple.correct())
         {
            // SOLVING FOR PRESSURE FIELD
            #include "pEqn.H"
         }

         // CALCULATION OF FLUID PROPERTIES BASED ON NEW TEMPERATURE
         #include "variableproperties.H"

         // CALCULATION OF SATURATION PRESSURE BASED ON NEW TEMPERATURE
         #include "calcPSatField.H"
      }

      // CALCULATION OF MIXTURE PROPERTIES
      k = 1.0 / ( (alpha1/k1) + (alpha2/k2) );
      nu = alpha1*nu1 + alpha2*nu2 ;
      cp = alpha1*cp1 + alpha2*cp2;
      rho = alpha1*rho1 + alpha2*rho2;

      forAll(alpha1, celli)
      {
         if (alpha1[celli] >= 0.5 )
         {
            liq += scalar(1);
            }
            else
            {
                vap += scalar(1);
            }
        }

        ratio = liq/vap;
        Info<<"liquid="<<liq<<"vapor="<<vap<<"liquid/vapor="<<ratio<<endl;

        liq = 0;
        vap = 0;

        Info<<"Min(p)="<<min(p).value()<<"Max(p)="<<max(p).value()<<endl;
        Info<<"Min(pSat)="<<min(pSat).value()<<"Max(pSat)="<<max(pSat).value()<<endl;
        Info<<"Min(alpha1)="<<min(alpha1).value()<<"Max(alpha1)="<<max(alpha1).value()<<endl;
        Info<<"Min(rho)="<<min(rho).value()<<"Max(rho)="<<max(rho).value()<<endl;
        Info<<"Min(T)="<<min(T).value()<<"Max(T)="<<max(T).value()<<endl;
        Info<<"Min(Tbdy)="<<min(Tbdy).value()<<"Max(Tbdy)="<<max(Tbdy).value()<<endl;

        runTime.write();
        Info<<"ExecutionTime="<<runTime.elapsedCpuTime()<<"s"<<"ClockTime="<<runTime.elapsedClockTime()<<"s"<<nl<<endl;
    }
    Info<<"End\n"<<   endl;
    return 0;
}

// ********************************************************************* //
```

Appendix E

Derivation of Clausius-Clapeyron Relation

The Clausius-Clapeyron Relation is going to be derived in this Appendix . It is defined as the slope of the tangents to the coexistence curve, the line separating the two phases of matter. Mathematically it can be stated as:

$$\frac{dp}{dT} = \frac{\mathit{\Delta} H}{T \mathit{\Delta} v} \tag{E.1}$$

where
$\mathit{\Delta} H$ = Specific Latent Heat
$\mathit{\Delta} v$ = Specific Volume Change of Phase Transition

Assuming that the evaporation of the liquid occurs at temperatures much lower that the critical temperature, the ideal gas approximation can be assumed which would imply that the specific volume of vapor is much greater that the specific volume of liquid ($v_v >> v_l$). So we assume $\mathit{\Delta} v \approx v_v$ and the Clausius Clapeyron relation for evaporation becomes:

$$\frac{dp}{dT} = \frac{\mathit{\Delta} H_v}{T v_v} \tag{E.2}$$

Using the Ideal Gas Law ($v_v = RT/p$) we can write:

$$\frac{dp}{dT} = \frac{\mathit{\Delta} H_v \; p}{T^2 R} \tag{E.3}$$

Generally the latent heat of vaporization varies along the coexistence curve as a function of temperature. However, if it is assumed to be a constant we can write:

$$\frac{dp}{p} = \left(\frac{\mathit{\Delta} H_v}{R} \right) \frac{dT}{T^2} \tag{E.4}$$

Integrating we obtain the Clausius-Claperon Relation

$$p = \exp \left[- \left(\frac{\mathit{\Delta} H_v}{R} \right) \frac{1}{T} + A \right] \tag{E.5}$$

Appendix F

Expansion of Terms

In this Appendix, expansion of some frequently used term is going to be carried out. a, b and c are considered scalars while $\boldsymbol{V}$ is a vector.

Expansion of Convection Term

$$\nabla \cdot (abc\boldsymbol{V}) = \left(\frac{\partial}{\partial x}\hat{\imath} + \frac{\partial}{\partial y}\hat{\jmath} + \frac{\partial}{\partial z}\hat{k}\right) \cdot (abc\boldsymbol{V_x}\hat{\imath} + abc\boldsymbol{V_y}\hat{\jmath} + abc\boldsymbol{V_z}\hat{k}) \tag{F.1}$$

The expansion of the convection term leads to:

$$\begin{aligned}
\nabla \cdot (abc\boldsymbol{V}) &= \frac{\partial}{\partial x}(abc\boldsymbol{V_x}) + \frac{\partial}{\partial y}(abc\boldsymbol{V_y}) + \frac{\partial}{\partial z}(abc\boldsymbol{V_z}) \\
&= a\frac{\partial}{\partial x}(bc\boldsymbol{V_x}) + bc\boldsymbol{V_x}\frac{\partial a}{\partial x} + a\frac{\partial}{\partial y}(bc\boldsymbol{V_y}) + bc\boldsymbol{V_y}\frac{\partial a}{\partial y} \\
&\quad + a\frac{\partial}{\partial z}(bc\boldsymbol{V_z}) + bc\boldsymbol{V_z}\frac{\partial a}{\partial z} \\
&= a\left[b\frac{\partial}{\partial x}(c\boldsymbol{V_x}) + c\boldsymbol{V_x}\frac{\partial b}{\partial x}\right] + bc\boldsymbol{V_x}\frac{\partial a}{\partial x} + a\left[b\frac{\partial}{\partial y}(c\boldsymbol{V_y}) + c\boldsymbol{V_y}\frac{\partial b}{\partial y}\right] \\
&\quad + bc\boldsymbol{V_y}\frac{\partial a}{\partial y} + a\left[b\frac{\partial}{\partial z}(c\boldsymbol{V_z}) + c\boldsymbol{V_z}\frac{\partial b}{\partial z}\right] + bc\boldsymbol{V_z}\frac{\partial a}{\partial z} \\
&= bc\left[\boldsymbol{V_x}\frac{\partial a}{\partial x} + \boldsymbol{V_y}\frac{\partial a}{\partial y} + \boldsymbol{V_z}\frac{\partial a}{\partial z}\right] + ac\left[\boldsymbol{V_x}\frac{\partial b}{\partial x} + \boldsymbol{V_y}\frac{\partial b}{\partial y} + \boldsymbol{V_z}\frac{\partial b}{\partial z}\right] \\
&\quad + ab\left[\frac{\partial}{\partial x}(c\boldsymbol{V_x}) + \frac{\partial}{\partial y}(c\boldsymbol{V_y}) + \frac{\partial}{\partial z}(c\boldsymbol{V_z})\right]
\end{aligned} \tag{F.2}$$

So the convection term can be written as:

$$\nabla \cdot (abc\boldsymbol{V}) = bc\,(\boldsymbol{V} \cdot \nabla a) + ac\,(\boldsymbol{V} \cdot \nabla b) + ab\,[\nabla \cdot (c\boldsymbol{V})] \tag{F.3}$$

Expansion of Temporal Derivative

$$\begin{aligned}\frac{\partial}{\partial t}(abc) &= a\frac{\partial}{\partial t}(bc) + bc\frac{\partial a}{\partial t}\\ &= a\left[b\frac{\partial c}{\partial t} + c\frac{\partial b}{\partial t}\right] + bc\frac{\partial a}{\partial t}\\ &= ab\frac{\partial c}{\partial t} + ac\frac{\partial b}{\partial t} + bc\frac{\partial a}{\partial t}\end{aligned} \tag{F.4}$$

The expansion yields:

$$\frac{\partial}{\partial t}(abc) = ab\frac{\partial c}{\partial t} + ac\frac{\partial b}{\partial t} + bc\frac{\partial a}{\partial t} \tag{F.5}$$

Bibliography

[1] Payam Rahimi, Charles A. Ward, "Kinetics of Evaporation: Statistical rate Theory Approach", International Journal of Thermodynamics, Vol. 8(No.1), 2005, pp. 1-14.

[2] J.T. Zhang and B.X. Wang, "Effect of Capillary at Liquid-Vapor Interface on Phase Change without Surfactant", International Journal of Heat and Mass Transfer 45, 2002, pp. 2689-2694.

[3] Martin Wörner, "Numerical Modelling of Multiphase Flows in Microfluids and Micro Process Engineering: A Review of Methods and Applications", Microfluid Nanofluid 12, 2012 pp. 841-886.

[4] James M. Hyman, "Numerical Methods for Tracking Interfaces", Physica 12D, 1984, pp. 396-407.

[5] Onnu Ubbink, "Numerical Prediction of two fluid systems with Sharp Interfaces", Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, 1997.

[6] C. W. Hirt and B. D. Nichols, "Calculating Three Dimensional Free Surface Flows in the Viscinity of Submerged and Exposed Structures", Journal of Computational Physics, Vol.12, 1973, pp. 234-246.

[7] Osher S. and Sethian, J.A, "Fronts Propagating with Curvature-Dependent Speed: Algorithms based on Hamilton-Jacobi Formulations", Journal of Computational Physics Vol.79, 1988, pp. 12-49.

[8] Francis H. Harlow and J. Eddie Welch, "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface", The Physics of Fluids, Vol.8 (No.12), 1965, pp. 2182-2189.

[9] C. W. Hirt and B. D. Nichols, "Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries", Journal of Computational Physics 39, 1981, pp. 201-225.

[10] Hrvoje Jasak, "Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows", Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, 1996.

[11] H.G. Weller et al., "A Tensorial Approach to Computational Continuum Mechanics using Object-Oriented Techniques", Computers in Physics Vol. 12, 1998.

[12] Raees et al., "Evaluation of the Interface Capturing Algorithm of OpenFOAM for the Simulation of Incompressible Immiscible Two Phase Flow", Reports of the Department of Applied Mathematical Analysis, Delft University of Technology, 2011.

[13] Samuel W. J. Welch and John Wilson, "A Volume of Fluid Based Method for Fluid Flows with Phase Change", Journal of Computational Physics 160, 2000, pp. 662-682.

[14] S. Mostafa Ghiaasiaan, "Convective Heat and Mass Transfer", ISBN 978-1-107-00350-7, 2011, pp. 24-26.

[15] V.K. Badam et al., "Experimental and Theoretical Investigations on Interfacial Temperature Jumps during Evaporation", Experimental Thermal and Fluid Sciences 32, 2007, pp. 276-292.

[16] Van P. Carey, "Liquid-Vapor Phase Change Phenomena", ISBN 0-89116836, 1992, pp. 112-121.

[17] The OpenFOAM Foundation, http://www.openfoam.com

[18] Rolf Krahl et al., "A Model for Two Phase Flow with Evaporation", Weierstrass Institute for Applied Analysis and Stochastics, 2004.

[19] Herve Lemonnier, Didier Jamet and Olivier Lebaigue, "Validation of Advanced Computational Methods for Multiphase Flow", ISBN 1-56700-218-8, 2005, pp. 45-63.

[20] Diana Gaulke and Michael Dreyer, "A Study of the accommodation Coefficient in Flow3D", ZARM, Monza, 2011.

[21] OpenFOAM Foundation, "OpenFOAM Programmer's Guide", Version 2.2.0, 2013.

[22] Suhas V. Patankar, "Numerical Heat Transfer and Fluid Flow", ISBN 0-07-048740-5, 1980, pp. 44-46.

[23] 3M Speciality Materials, "3M Novec^{TM} Engineered Fluid HFE-7000 Physical Properties", www.3m.com/fluids.

[24] R. Brodkey and H. Hershey, "Transport Phenomena - A Unified Approach, Volume 2, Part III - Transport Property", McGraw-Hill Chemical Engineering Series.