7-2018

Numerical Investigation of Heat and Mass Transfer Phenomena in Boiling

Isaac Bernabe Perez-Raya
ibp1401@rit.edu

R·I·T

Numerical Investigation of Heat and Mass Transfer Phenomena in Boiling

by

Isaac Bernabe Perez-Raya

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctorate of Philosophy in Microsystems Engineering

Microsystems Engineering Program
Kate Gleason College of Engineering

Rochester Institute of Technology
Rochester, New York
July, 2018

Numerical Investigation of Heat and Mass Transfer Phenomena in Boiling

by

Isaac Bernabe Perez Raya

Committee Approval:

We, the undersigned committee members, certify that we have advised and/or supervised the candidate on the work described in this dissertation. We further certify that we have reviewed the dissertation manuscript and approve it in partial fulfillment of the requirements of the degree of Doctor of Philosophy in Microsystems Engineering.

Dr. Satish G. Kandlikar — Date
Professor, Mechanical Engineering

Dr. Steven Day — Date
Associate Professor, Biomedical Engineering

Dr. Kara L. Maki — Date
Associate Professor, School of Mathematical Sciences

Dr. Ivan Puchades — Date
Assistant Professor, Electrical and Microelectronic Engineering

---

Dr. Jiandi Wan — Date
Assistant Professor, Microsystems Engineering

Certified by:

Dr. Bruce Smith — Date
Director, Microsystems Engineering

Contents

List of Figures

List of Tables

Nomenclature

A	Hamaker constant (J)
A_i	Interface surface area (m^2)
C	Large coefficient
c	Specific heat (J/kg-K)
d	Distance Interface to cell center (m)
F	Volume fraction
f	Evaporation coefficient
g	Gravitational Constant (m/s^2)
h_{fg}	Latent heat (J/kg)
K	Curvature (1/m)
k	Thermal conductivity (W/m-K)
M	Molecular weight (kg/mol)
m''	Mass flux (kg/s-m^2)
$\vec{n}$	Unit normal vector
P_l	Liquid pressure in microlayer (Pa)
p	Pressure (Pa)
$\dot{Q}$	Integrated heat flux (W/m)
q''	Heat flux (W/m^2)

R Universal gas constant (J/mol-K)

R_b Bubble radius (m)

R_{int} Interfacial resistance (m^2-K/W)

r Radial coordinate (m)

S_M Momentum source term (N/m^3)

S_T Energy source term (K/s)

S_{mic} Mass source term evaporation (kg/s-m^3)

s_d Distance interface to cell center (m)

T Temperature (K)

t Time (K)

u Radial or Cartesian x velocity (m/s)

V Volume (m/s)

$V_{m,l}$ Liquid molar volume (m^3/mol)

v Fluid velocity (m/s)

X Interface position (m)

x Cartesian x coordinate (m)

z Axial coordinate (m)

Greek Letters

α Thermal diffusivity (m^2/s)

β Interface displacement constant (m^2/s)

δ Microlayer thickness (m)

ϵ Density ratio

η Microlayer axial coordinate (m)

Γ Mass flux per unit contact line length (kg/s-m)

μ Dynamic viscosity (Pa-s)

ν Kinematic viscosity (m^2/s)

ρ Density (kg/m^3)

σ Surface tension (N/m)

θ Angle from vertical boundary (rad)

φ Apparent contact angle (rad)

ξ Microlayer radial coordinate (m)

Subscripts

ad Adsorbed film

cl Contact line

l Liquid

lv Liquid-vapor interface

mic Microlayer

sat Saturation

v Vapor

w Wall

Acknowledgments

I would like to thank Professor Kandlikar for the opportunity to work in his Lab. His enthusiasm and passion were a true motivator for me to be a better person and researcher. I am grateful that he introduced me to the leaders in the field of numerical heat transfer and other research areas. The world that he showed me has become a pillar in my desire to be a researcher.

I want to thank the members of my dissertation committee for their time, patience, and intellectual support. I am in debt to Dr. Steven Day for his suggestions on the development of my project. To Dr. Kara Maki for her support and advice on analyzing fundamental problems of phase-change and numerical methods. To Dr. Jiandi Wan for being open to discuss my ideas and for asking me questions that challenged me to think at a deeper level.

To my lab-mates thanks for the fun and support. Mustafa and Rupak gave me suggestions to be a better Ph.D. student. Ankit, Arvind, and Pruthvik helped me to improve my technical presentations. Thanks also to Alyssa, Aniket, Aranya, Jose Luis, and Travis for their friendship. My special thanks to Simon Shipkowski for the discussions that we kept related to interface reconstruction and other important topics.

I would like to thank my wife Lorena (Lore) for her support and infinite love. She gave me words of encouragement at critical moments during the development of my Ph.D. project. Thank you Lore for your support on pursuing my dream of becoming a doctor. I am sorry for all the Sundays that I decided to go back to the lab instead of spending time with you. I love you Lore!

Finally, but not least, I want to thank God for his unconditional love. He believed that I could develop my Ph.D. project at RIT. Moreover, He gave me the opportunity to get closer to him during these last years.

Abstract

Applications boiling are found in heat sinks for electronics cooling, nuclear and fossil fuel powered steam generators, distillation columns, concentrated solar power systems, glass melting furnaces, desalination chambers, and heat and mass exchangers. In order to increase the performance and safety margins of these applications, there is a need to develop tools that predict the thermal and fluid behavior during bubble growth. The analysis of boiling has been addressed by computer simulations, which employ methods for approximating mass and heat transfer at the interface.

However, most simulations make assumptions that could adversely affect the prediction of the thermal and dynamic fluid behavior near the bubble-edge. These assumptions include: (i) computation of mass transfer with local temperature differences or with temperature gradients at cell-centers rather than with temperature gradients at the interface, (ii) modified discretization schemes at neighboring-cells or a transition region to account for the interface saturation temperature, and (iii) interface smearing or distribution of mass transfer into multiple cells around the interface to prevent interface deformations.

The present work proposes methods to perform a simulation of nucleate boiling. The proposed methods compute mass transfer with temperature gradients at the interface, account for the interface saturation temperature, and model a sharp interface (interface within one cell) with mass transfer only at interface-cells. The proposed methods lead to a more realistic representation of the heat and mass transfer at the interface. Results of the simulation are in excellent agreement with theory on planar interface evaporation and growth of spherical bubbles in superheated liquid. In addition, numerical bubble growth rates compare well with experimental data on single bubble nucleation over a heated surface. The simulation of nucleate boiling with water and a 6.2 K wall superheat reveals large heat transfer coefficients over a 200 μm distance from the interface. In addition, analyses of the wall shear stress indicate an influence region of two-times the departure bubble diameter.

1 Introduction

1.1 Background

Heat dissipation is a critical aspect of state-of-the-art technologies within many applications including space, military, and industry. For instance, spaceflight vehicles require effective heat removal techniques to operate in extreme environments. Military applications use small-area powerful electronic devices (e.g. processor chips, laser diodes) that require dissipation of high heat fluxes ($>10^3$ W/cm^2). Industry uses refrigeration systems to reach temperatures below -155 °C (-247 °F), needed for conservation and production of food, pharmaceuticals, and ethylene or liquefied natural gas. Moreover, components in nuclear reactors (e.g. limiters, diverters, and beam dumps) require cooling of ultra-high heat fluxes (10^3-10^5 W/cm^2) in a continuous manner.

Boiling dissipates large amounts of heat while keeping a low and a constant surface temperature. Therefore, boiling appears in many industrial applications as a heat removal technique. Other systems where boiling appears include steam generators in nuclear and fossil fuel plants, distillation columns, concentrated solar power systems, glass melting furnaces, and desalination chambers. However, to be able to apply boiling in state-of-the-art technologies, we need to understand the way bubbles form and its relationship with the surface and surrounding liquid.

Sophisticated experiments with interferometry, Micro-Electro-Mechanical-Systems (MEMS), and high-speed photography, reveal information on the heat and flow transport during boiling. However, these experiments are limited by their resolution and ranges of operation. For instance, MEMS can measure temperature with a maximum grid resolution of 20 μm, which is large relative to the microlayer length. In addition, it is challenging and expensive to perform these experiments at extreme environments such as microgravity, high pressures, or low temperatures.

Computer simulations provide detailed information on the dynamic and thermal behavior

of the fluid during boiling while showing good agreement with experimental data [1]. In addition, simulations of boiling determine the contribution of microlayer effects to the overall heat and mass transfer. Another advantage is the possibility of analyzing the behavior of boiling at critical conditions just by changing the values of the parameters in the simulation.

However, the simulation of boiling is a nontrivial task due to the multiple physics involved including: (i) mass transfer at the bubble-edge that depends on interface inclination and fluid temperature near the interface, (ii) the heat transfer at the interface, and (iii) the conditions at the interface such as surface tension. Accurate methods available in the technical literature to simulate bubble growth require advanced algorithms that present significant challenges in the analysis of phase change at the interface level. Other approaches require simplifying assumptions at the expense of accuracy.

The objective of this study is to develop numerical methods to perform an accurate and robust simulation of the heat and mass transfer processes in boiling. The models should simulate bubble growth with a sharp interface to predict the dynamic and thermal fluid behavior near the interface. Such information is important to gain a better understanding of the heat transfer mechanisms in boiling, which is needed to propose techniques that predict and improve the performance of boiling at various operating conditions.

1.2 Previous Related Studies

1.2.1 Mass Transfer Models

Figure 1(a) shows a growing bubble immersed in liquid in contact with a heated plate. The bubble grows due to evaporation of liquid at the interface; the amount of liquid that evaporates at the interface is termed interfacial mass flux ($m_e{}''$). The interfacial mass flux is proportional to the heat flux ($q_e{}''$) that arrives to the interface. In addition, the interfacial heat flux depends on the magnitude of the temperature gradient at the interface in a direction normal to the interface ($\frac{\partial T}{\partial n}_{int}$). Eq. (1) states the dependence between the mass flux, the

heat flux, and the interfacial temperature gradients:

$$m_e{}'' = \frac{q_e{}''}{h_{fg}} = \frac{k_l}{h_{fg}} \frac{\partial T}{\partial n}_{int}, \tag{1}$$

where k_l is the liquid thermal conductivity, and h_{fg} is the latent heat of evaporation.

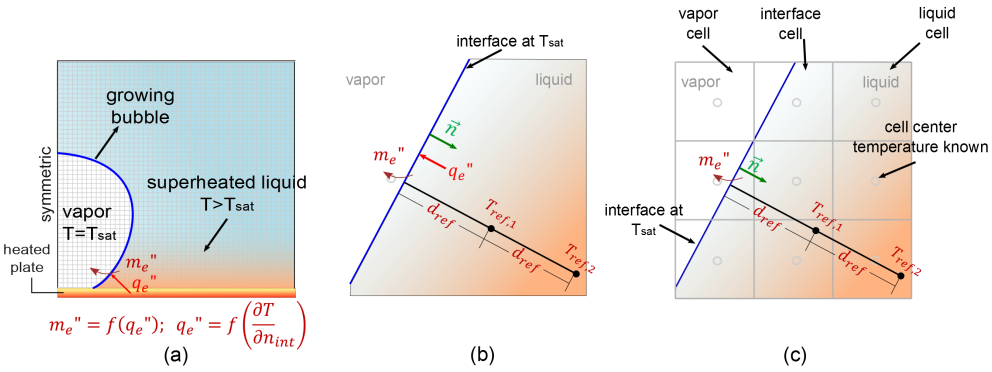

Figure 1: Need to compute mass transfer in a simulation of boiling: (a) computational domain with a bubble, (b) estimation of temperature gradients at the interface, and (c) estimation of temperature gradients in computational cells.

The estimation of the temperature gradients at the interface in a normal direction requires temperatures at reference points located in a direction normal to the interface (see Figure 1(b)). Some options to calculate the interfacial temperature gradients include first or second order approximations; a first order approximation considers the temperature of only one reference point on the liquid, and a second order approximation considers the temperature of two reference points on the liquid. Eqs. (2)-(3) show the formula to estimate the temperature gradient with first and second order approximations:

$$\frac{\partial T}{\partial n}_{int,1st} = \frac{T_{ref,1} - T_{sat}}{d_{ref}} \tag{2}$$

$$\frac{\partial T}{\partial n}_{int,2nd} = \frac{-T_{ref,2} + 4T_{ref,1} - 3T_{sat}}{2d_{ref}}, \tag{3}$$

where T_{sat} is the saturation temperature at the interface, $T_{ref,1}$ is the reference temperature at a distance d_{ref} from the interface along the normal direction, and $T_{ref,2}$ is the reference temperature at a distance $2d_{ref}$ from the interface along the normal direction (see Figure

1(b)).

The numerical simulation divides the computational domain into multiple cells (see Figure 1(c)). These computational cells are classified as vapor-cells (cells filled with vapor), liquid-cells (cells filled with liquid), or interface-cells (cells that have an interface). The cells that lie next to interface-cells are known as neighboring-cells. The simulation should estimate the mass transfer that corresponds to each interface-cell. In addition, the numerical model solves the governing equations in a way that it finds the temperature at the cell centers. However, the location of the cell centers differs from the location of the reference points along the interface normal direction. Therefore, the computation of the mass transfer requires models or methods to determine the temperatures at the reference locations by using the temperature at the computational cell centers.

The technical literature distinguishes various models to estimate interfacial temperature gradients in computer simulations. Figure 2 shows common methods to estimate the interfacial temperature gradients. Figure 2(a) shows the multiple-cells method proposed by Udaykumar and Shyy [2], Figure 2(b) shows the neighboring-cell method proposed by Sato and Niceno [3], and Figure 2(c) shows the interfacial-resistance method [4].

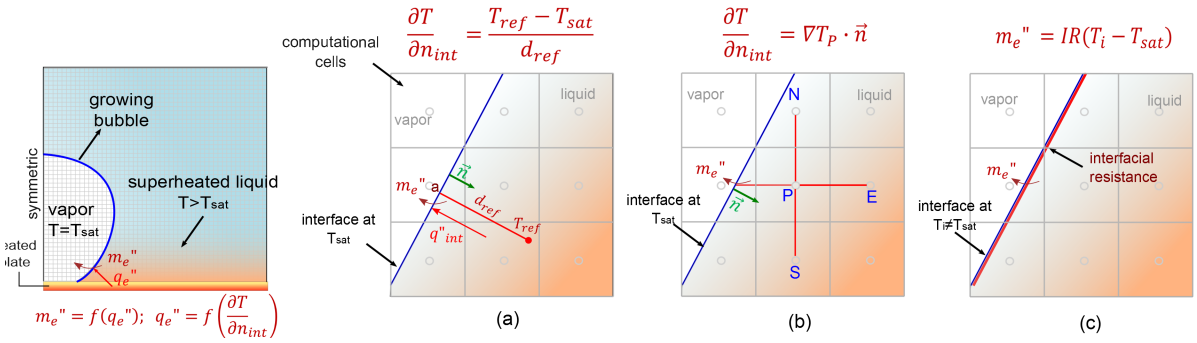

Figure 2: Methods to compute the temperature gradients approximated by: (a) multiple-cells method, (b) neighboring-cell method, and (c) interfacial-resistance method.

Figure 2(a) shows the multiple-cells method proposed by Udaykumar and Shyy [2]. The multiple-cells method uses a biquadratic function to interpolate the temperature of a reference point located in a direction normal to the interface. The temperature at the center of

multiple-cells (five cells) around the reference point and the temperature of the interface at point "a" (given by the normal direction from the center of the interface-cells, see Figure 2(a)) determine the coefficients of the interpolation function. Retrieving the temperature of multiple cells along interfaces that continuously change its orientation requires advanced numerical procedures; therefore, simulations of nucleate boiling prefer to use less accurate models. Nevertheless, a few simulations of film boiling and rising bubbles in a pool of liquid have used the multiple-cells method [5, 6].

Other methods compute the temperature gradients at the center of the cells to avoid the interpolation of temperatures at reference points [3,7]. Consider for instance the neighboring-cell method proposed by Sato and Niceno [3] (see Figure 2(b)). The neighboring-cell method assumes that the normal temperature gradient at the interface is equal to the temperature gradient at the neighboring cell center along Cartesian directions times the normal vector. The neighboring-cell method computes the temperature gradient based on the temperature of the surrounding cells (e.g. cells N, S, E, in the figure) and the temperature and location of the interface. When the interface does not intersect the line that connects the centers of the interface-cell and the neighboring-cells, the temperature gradient uses conventional discretization schemes that ignore the interface. The method requires the temperature of three or four computational cells and estimates the temperature gradient at the center of the neighboring-cell rather than at the interface. Therefore, the accuracy of the method suffers when the interface is far from the neighboring-cell center.

The interfacial-resistance model assumes an interfacial resistance at the interface and that the temperature of the interface differs from the saturation temperature (see Figure 2(d)). The interfacial-resistance model computes mass transfer without the temperature of surrounding liquid-cells. Juric and Tryggvason [8] simulated film boiling with the interfacial-resistnace model; the authors made an entropy balance across the interface to find the temperature of the interface. Various simulations of boiling have used the interfacial resistance-model [9–15]. However, as pointed out by Hardt and Wondra [9], the interfacial-resistance

model applies to cases where the interface lies at a submicron distance from a surface, and errors may appear as the distance becomes larger.

1.2.2 Methods for Interface Temperature

Figure 3(a) shows a growing bubble immersed in liquid in contact with a heated plate. The transformation of liquid into vapor at the interface occurs at the saturation temperature, which implies that the fluids in contact with the interface are at the saturation temperature. Figure 3(b) shows a diagram of the parameters involved in the modeling of the interface temperature. The simulation should ensure that the interface-cells and the liquid-neighboring-cells see the interface temperature.

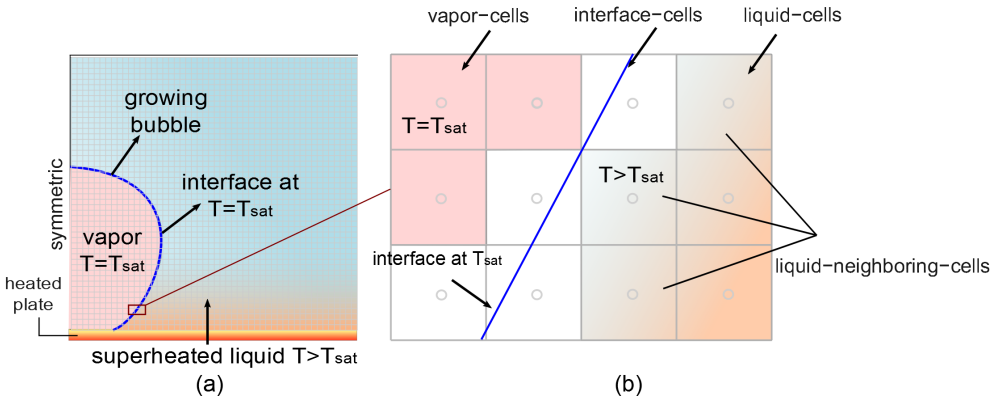

Figure 3: Need to include the interface temperature: (a) nucleating bubble with an interface at the saturation temperature, (b) computational cells with an interface.

Figure 4 shows common numerical techniques to model the interface. These methods include the assumption of a transition region near the interface [16–24] (see Figure 4(a)), modified neighboring-cells that use the interface as a boundary [3, 25–27] (see Figure 4(b)), and fixing the temperature of the interface-cells by considering the interface temperature and location (ghost fluid method) [7, 28–35] (see Figure 4(c)).

Son et al. [17] used a function to smooth the liquid thermal conductivity on a transition region (three cells thick) around the interface and imposed the saturation temperature on the vapor-cells that lie outside the transition region (see Figure 4(a)). A Heaviside func-

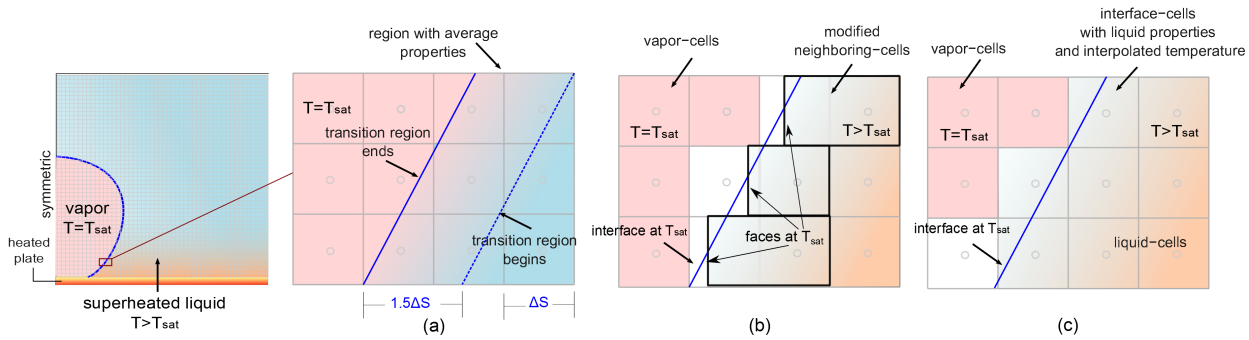

Figure 4: Methods to include the interface temperature in the simulation: (a) transition-region, (b) modified neighboring-cells, (c) ghost-fluid.

tion defined the transition region by using information on the Level-Set interface tracking algorithm (level-set values indicating the distance from the cell center to the interface). The transition region method ignores the exact interface location and temperature, which may lead to errors on imposing the interface temperature condition as stated by Son and Dhir [28].

The modified neighboring-cells method modifies the discretization schemes at the neighboring-cells. The modification moves the face (or boundary) of the neighboring-cell to the interface location and assigns the saturation temperature at the relocated face (see Figure 4(b)). Sato and Niceno [3] used second order discretization schemes to determine the temperature gradient at the neighboring-cell center based on the interface location. The modified neighboring-cells method is accurate since it considers the exact interface location and temperature. However, such a method works only for interfaces that cross the line that connects the centers of the interface-cells and the neighboring-cells (centerline). Interface-cells with an interface that does not cross the centerline used conventional discretization schemes that ignore the interface effects. In the case of using a numerical package to perform the simulation, the method requires the elimination of the software procedures at the neighboring-cells, and the integration of the modified discretization schemes, which is a complex task in 2D and 3D simulations.

The ghost-fluid method assumes that the interface-cells have liquid properties (see Figure 4(c)). In addition, the ghost-fluid method interpolates the temperature of the interface-cells

by using the temperature of the liquid cells and the temperature of the interface. The method is efficient since it requires conventional discretization schemes on the neighboring-cells (the neighboring-cells see the interface by reading the interpolated interface-cell temperature); however, the method depends on an interpolation function. Gibou et al. [29,30] constructed first and second order approximations to interpolate the temperature of the interface-cells. Temperature gradients along Cartesian directions were used to perform the interpolation. Ling et al. [7] found the temperautre of the interface-cells with a mathematical expression. The expression assumed that the temperature gradient at the interface was equal to the temperature gradient at the interface-cell center. This assumption is valid as long as the interface is near the interface-cell center. Another alternative is to use a partial differential equation approach to interpolate the temperature of the interface-cells [31]. This approach considers a hyperbolic partial-differential-equation (PDE) that performs the extrapolation by considering the vector normal to the interface and the derivatives of the extrapolated variable. Tanguy et al. [32] considered a linear extrapolation with the PDE approach to find the temperatures at the interface-cells. The method requires the solution of an extra PDE and the communication of the new PDE with the governing equations.

1.2.3 Tracking Interface Models

In boiling, the bubble-edge is a sharp division between vapor and liquid phases. The bubble-edge dynamically changes due to the conversion of liquid into vapor. In addition, buoyancy and surface tension forces contribute to the bubble-edge motion; buoyancy forces lift the bubble from the surface, whereas surface tension holds the bubble to the surface [36].

Simulations of boiling depend on interface-tracking algorithms. The interface-tracking algorithm uses the velocities at the interface and the interfacial mass transfer to estimate the interface displacement at each time step. In addition, the simulation uses the interfacial gradients of the interface tracking variable to compute surface tension effects. The technical literature distinguishes various interface-tracking algorithms. Four of the most common

algorithms applied to simulate boiling are: (i) particles on interface method, (ii) Level-Set (LS) method, (iii) Volume-of-Fluid (VOF) method, and (iv) Coupled Level-Set and Volume-of-Fluid (CLSVOF) method.

Daly [37] introduced the particles on interface method, which tracks the interface with marker particles that change its location based on the fluid velocity. The method joins the particles to determine the orientation of the interface, which leads to precise estimations of interface curvature and surface tension effects. The method requires the elimination or inclusion of particles as the interface evolves. One serious disadvantage is that surface tension errors might appear due to fluctuations in the markers position when the markers are too close. Udaykumar and Shyy [2] simulated melting of gallium with a lagrangian translation of marker particles to track the interface. The interfacial mass flux (given by temperature gradients at the interface) determined the velocity of the markers. Juric and Tryggvason [8] adopted a similar interface tracking algorithm to simulate film boiling; the simulation used an indicator function to define the fluid properties. Simulations of film boiling indicated that interface tracking with marker particles capture complex interfaces with abrupt changes in curvature while keeping proper fluid velocities (see Figure 5).

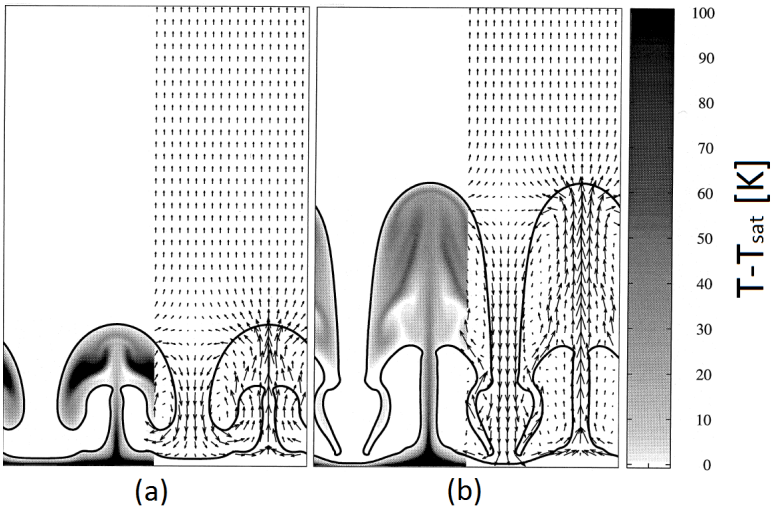

Figure 5: Interface evolve and temperature in the simulation of film boiling with marker particles at two different times. Fluid: hydrogen at 8 atm., (figure adapted from [8]).

Osher and Sethian [38] proposed the Level-set method (LS), which tracks the interface

with the variable ϕ that indicates the distance from the cell to the interface. A value of $\phi = 0$ corresponds to the interface position, $\phi > 0$ corresponds to phase-1, and $\phi < 0$ corresponds to phase-2. An advection equation changes the values of ϕ based on the interface velocity. One advantage of the LS lies in the smooth transition of ϕ across the interface. The smooth transition leads to accurate estimation of the gradients of ϕ at the interface used to compute surface tension. Therefore, the LS method is capable of capturing multiphase flows where surface tension effects play a primary role (e.g. nucleate boiling). However, the method depends on a reinitialization procedure to ensure that ϕ remains accurate, which affects mass conservation. The technical literature distinguishes various works that use the LS method to track the interface [17, 19, 20, 34, 35]. Son et al. [17] simulated nucleate boiling with the LS method. The simulation used ϕ to define the transition region (three cells thick) around the interface to avoid numerical instabilities. The mass flux at the interface gave the interface velocity. Gradients of ϕ computed surface tension effects in the momentum equation. Results showed sharp interfaces without appreciable deformations throughout the ebullition cycle (see Figure 6). The interface remained sharp even near the contact line, which is a critical region due to the abrupt changes in curvature and strong evaporation.

The Volume-of-Fluid (VOF) method proposed by Hirt and Nichols [39] is mass conservative. The VOF method tracks the interface by using F_1, which represents to the volume-fraction of phase-1 at the computational cells. In the VOF method, cells with $F_1 = 1$ are phase-1 cells, cells with $F_1 = 0$ are phase-2 cells, and cells with $0 < F_1 < 1$ are interface-cells. An advection equation moves the interface based on the fluid velocities and mass source terms at the interface. The method is free of reinitialization procedures since F_1 changes just at the interface-cells (the values of F_1 at cells far from the interface remain constant). However, one of the main difficulties lies in the abrupt change of F_1 across the interface. This abrupt change makes it difficult to define a smooth transition across the interface, which is needed to include surface tension effects. In addition, the VOF method requires an interface reconstruction step to properly move the interface in 2D and 3D simulations. Welch and

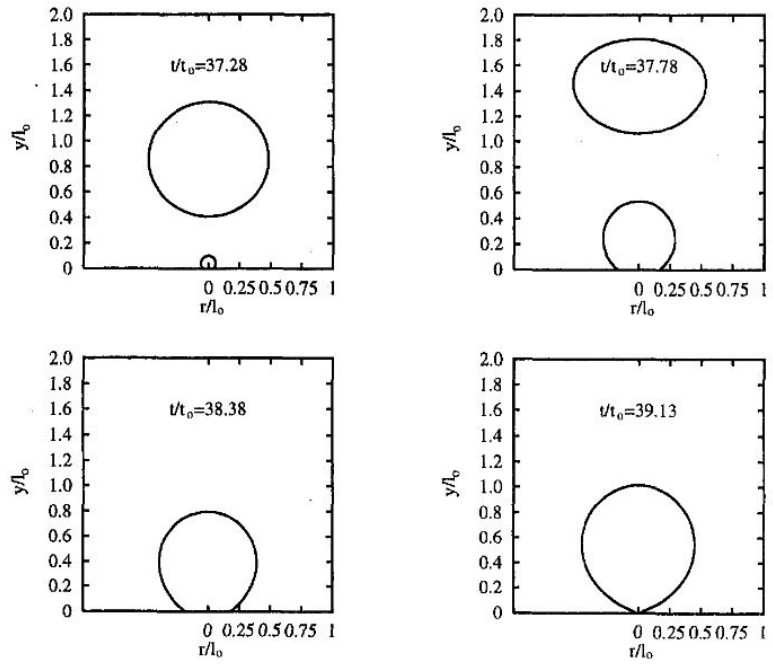

Figure 6: Interface dynamics in the simulation of nucleate boiling with the level set method. Fluid: water at 1 atm., $\Delta T_w = T_w - T_{sat} = 5$ K, $t_0 = 16$ ms, $l_0 = 2.5$ mm (figure adapted from [17]).

Wilson [40] were the first to use VOF to perform simulations of film boiling. The simulation assumed that the interface had a piecewise linear shape within each computational cell. Volume-of-fractions of nine cells around the interface-cell determined the gradients of volume-fractions in the estimation of the normal vector. Smoothed volume-fractions defined the transition region in the computation of surface tension. Results indicated that the VOF method is capable of handling complex interfaces in film boiling while keeping proper velocity distributions near the bubble-edge. A few other authors report the use of VOF to simulate nucleate boiling [3,10]. Kunkelmann and Stephan [10] smoothed the volume-fractions in the computation of surface tension. In addition, the authors smeared the mass source term over a few computational cells around the interface to avoid numerical instabilities, as proposed by Hardt and Wondra [9]. Sato and Niceno [3] tracked the interface with a color function ϕ that represents the volume-fraction of liquid inside the computational cell. Different than the traditional F_1, ϕ defines a transition region (three cells thick) around the interface and $\phi = 0.5$ defines the interface position. The method requires an interface sharpening equa-

tion to keep the interface sharp. Simulations of growth of spherical bubbles in uniformly superheated liquid revealed that the color function generates proper velocity jumps at the interface. However, nonphysical velocities appeared near the interface at low mass transfer rates that deformed the temperature field (see Figure 7). In addition, results indicated that the color function leads to proper bubble shapes in the simulation of nucleate boiling.

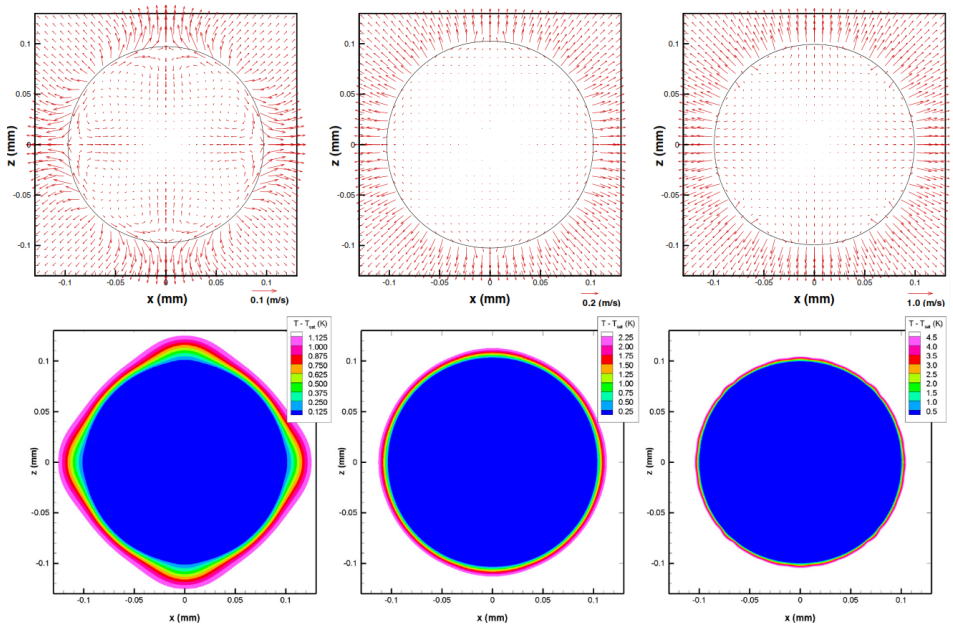

Figure 7: Velocity vectors and temperature distribution on spherical growing bubbles obtained with the VOF method and color function. $\Delta T = 1.25K$ (left), $\Delta T = 2.5K$ (middle), $\Delta T = 5.0K$ (right). Fluid: water 1 atm., (figure adapted from [3]).

The Coupled Level-Set and Volume-of-Fluid method (CLSVOF) proposed by Sussman and Puckett [41] combines the advantages of the LS and VOF method. The method uses the smooth transition of ϕ to compute surface tension and moves the interface with the VOF method to ensure mass conservation. However, the method is more complicated to implement since it requires the solution and coupling of two interface tracking algorithms [42]. Sun and Tao [43] reconstructed the level-set values with algorithms that detect the minimum distance from the cells to the interface. Ling et al. [7] simulated nucleate boiling with the CLSVOF; in the simulation, VOF moved the interface and defined the level-set values. A smooth Heaviside function defined a transition region at the interface based on the level-

set values. The Heaviside function defined fluid properties and gradients in the estimation of surface tension effects. Results indicated that the CLSVOF method captures interface evolution in nucleate boiling with proper velocities near the bubble-edge (see Figure 8). A similar CLSVOF method was adopted by Kunkelmann and Stephan [44] to simulate nucleate boiling.

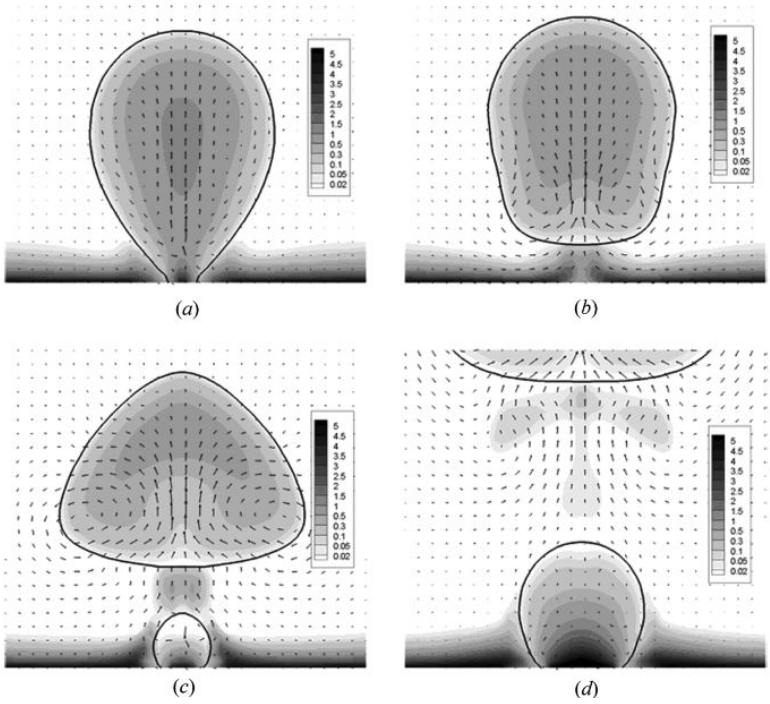

Figure 8: Velocity vectors and bubble shape obtained with the CLSVOF method in the simulation of nucleate boiling. (a) $t = 91.5$ ms, (b) $t = 97.5$ ms, (c) $t = 105$ ms, (d) $t = 132$ ms. Water 1 atm $\Delta T_w = 7K$ (figure adapted from [7]).

1.2.4 Contact Line Models

The region where the interface comes close to the wall is a key aspect of multiphase flows such as boiling. The interaction between the liquid molecules and the wall atoms give rise to adhesion forces that keep the liquid in contact with the surface. In multiphase flows, the adhesion forces experience a change when an interface approaches the heated wall, which results in a change in the liquid pressure (also known as disjoining pressure). The variation of the disjoining pressure changes the interface shape from an intrinsic meniscus into a flat non-

evaporating liquid film of molecular thickness. Therefore, the interface shape near the surface consist of three main sections (see Figure 9): (i) an evaporating meniscus (also known as macrolayer) where the film thickness is relatively large and where the flow behavior depends on the interface curvature and the mass transfer, (ii) an evaporating thin film (also known as microlayer) where the flow behavior depends on the capillary and disjoining pressure, and (iii) a flat thin film (also known as adsorbed layer) where strong adhesion forces prevent the evaporation of liquid. The interface forms an apparent contact angle φ with the surface, which is the angle of inclination of the interface at the meniscus region.

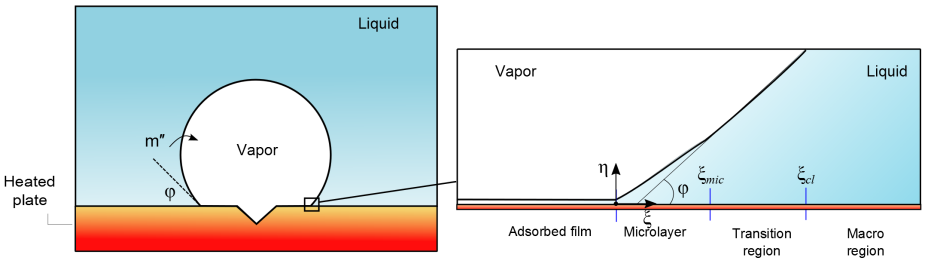

Figure 9: Representation of interface shape near the surface in nucleate boiling.

Wayner et al. [45] developed a theoretical model to determine the local heat transfer coefficient in the microlayer. The 1D momentum equation for laminar and fully developed flow determined the relation between the fluid velocity and the change in the liquid pressure. The boundary conditions of non-slip condition at the liquid-solid interface and zero shear-stress at the liquid-vapor interface gave the change in the liquid velocity along the axial direction $u(\eta)$. The integral of the liquid velocity along the microlayer thickness gave the evaporative mass flux. The expression proposed for Derjaguin and Zorin [46] defined the relation between disjoining pressure (p_d) and microlayer thickness (δ) as $p_d = -A/\delta$ where A is the Hamaker constant. A second expression used an extended kinetic theory to express the evaporative mass flux in terms of the interfacial temperature difference and pressure difference. The final expression gave the ratio of the heat transfer coefficient at $\delta \to \infty$ to the heat transfer in the microlayer as a function of the ratio of the layer film thickness and the adsorbed layer thickness. Results indicated a maximum microlayer thickness of 0.21 μm

and a microlayer length of 0.74 μm for CCl_4 on fused silica at 293 K and a superheat of 0.217 K with a heat flux of the order of 20 W/cm^2.

Kamotani [47] analyzed the heat transfer coefficient of a liquid film in grooved heat pipes. Different to the formulation of Wayner et al. [45], the model assumed a 1D heat conduction through the microlayer to express the relation between the surface temperature, the saturation temperature, and the microlayer thickness. Such a relation and the evaporative mass flux from the momentum equation (as derived by Wayner [45]) gave a fourth order differential equation for the microlayer profile. In addition, the model used a mathematical definition to express the interface curvature in terms of the microlayer thickness. Numerical methods solved the differential equation. Results for aluminum grooved pipe filled with ammonia indicated a maximum microlayer thickness of 0.1 μm and a length of 0.3 μm; the microlayer heat transfer was 3.4 W/m whereas the total heat transfer was 6.9 W/m.

Stephan and Busse [48] analyzed the heat transfer on a liquid film and solid plate of a grooved heat pipe. The formulation of Wayner et al. [45] and Kamotani [47] defined the microlayer modeling. The 2D heat-conduction equation gave the heat transfer on the solid plate and on the liquid groove outside the microlayer. The fourth order differential equation derived by Kamotani [47] for the microlayer thickness was written as a system of four first order differential equations and integrated using Runge-Kutta method. An iterative procedure coupled the 2D model and the microlayer model; the iterations ended when the heat transfer through the wall into the microregion was equal to the heat transfer given by the microlayer model. Figure 10 shows results reported by Stephan and Busse [48]. Results revealed that interface curvature suffers a sudden change near the adsorbed layer. Such trend in the interface curvature induced a rapid change in the interface temperature from T_{sat} to T_{wall}. As a result, the heat flux increased to a maximum value of 5.3 kW/cm^2 near the adsorbed layer. The change in the interface temperature occurs due to the dependence between the interface temperature and the capillary pressure. Analyses of the heat transfer coefficient indicated that 45% of the total heat in the groove went through the microregion.

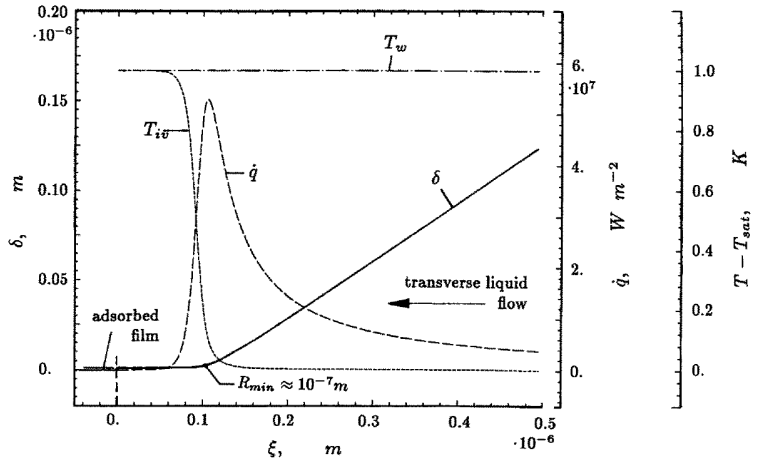

Figure 10: Liquid film shape, heat flux, and temperature in the microrgion (figure adapted from [48]).

Lay and Dhir [49] included the effect of recoil pressure (pressure gradient due to density difference) and gravity on the microlayer hydrodynamics and heat transfer. Bernoulli equation gave the variation of the vapor velocity and a force balance at the interface related the vapor velocity with the liquid velocity. An extended kinetic theory model expressed the interfacial heat flux as a function of the local temperature difference and pressure difference at the interface. The final expression was a fourth order differential equation for the microlayer. Results revealed that recoil pressure opposes the flow in the microlayer and that the effect of gravity is negligible. At 10 K superheat level in water, the microlayer length was 12 μm and 20 μm for a dispersion constant A of -1.1x10^{-21} J and 1x10^{-20} J, respectively. Larger wall superheats resulted in shorter microlayer lengths. Near the adsorbed layer, the heat flux had a maximum value of 15 kW/cm^2 at a 20 K superheat level.

Stephan and Hammer [50] performed a simulation of a bubble sited over a heated surface that included microlayer effects. The formulation of Wayner et al. [45] and Kamotani [47] modeled the microlayer. The 2D heat conduction equation gave the heat transfer on the solid plate and on the liquid outside the microlayer. The condition of similar curvature at the end of the microlayer and at the macroregion coupled the two models. Steady-state simulations of bubbles with different shape indicated the heat transfer during bubble growth

at various instances. Results for refrigerant R-114 and a copper plate indicated a maximum heat flux of 1.4 kW/cm^2 near the adsorbed layer. In addition, results revealed that about 38% of the total heat flow is due to microlayer effects covering only 0.16% of the copper plate. A comparison between numerical results and experimental data on the heat transfer coefficients at various heat fluxes indicated deviations of less than 8%.

Son et al. [17] were the first on performing a simulation of the complete ebullition cycle of a nucleating bubble that included microlayer effects. The level-set method tracked the interface in the macroregion. The model of Lay and Dhir [49] accounted for microlayer effects. A source term in the mass conservation equation at the contact line in the macrosimulation included microlayer evaporation. The mass source term represented the integral of the change in the mass flux along the microlayer length divided by the volume of the microlayer. The simulation defined the microlayer length as the location where microlayer thickness was equal to the distance from the first computational cell center to the heated surface. The microlayer length was 15 μm and the microlayer thickness was 12 μm. Results of water for wall superheats ($\Delta T_w = T_w - T_{sat}$) of 6.2, 7, and 8.5 K showed excellent agreement with experimental data in terms of bubble growth rates. Results revealed that the contribution of microlayer to the total heat flux is about 20%.

Kunkelmann and Stephan [10] coupled microlayer effects to a simulation of nucleate boiling with Volume-of-Fluid (VOF) interface tracking algorithm. The model of Stephan and Busse [48] estimated the heat flux and evaporation rate in the microlayer. The microlayer length was 0.5 μm and a transition region coupled the microlayer to the meniscus region. The transition region estimated the heat flux by assuming negligible effects of molecular forces on interface curvature. The simulation considered the heat transfer in the heated solid plate. Simulations with HFE-7100 at 500 mbar (0.5 atm) and a wall superheat of 5 K showed sharp temperature drops of the oder of 1 K at the contact line. However, the work simulated only one bubble cycle and a comparison with experimental data was not provided. Kunkelmann and Stephan [44] simulated multiple cycles of a nucleating bubble

to achieve periodic bubble growth regime. Numerical departure times and bubble diameters were 10% above experimental data. Figure 11 shows the variation of the heat flux over the surface reported by the authors; the results revealed peaks of heat flux of the order of 150 kW/m^2 at the contact line, and an increase in the heat flux during the advancing contact line (rewetting during detachment) due to colder liquid reaching the surface (transient conduction mechanism).

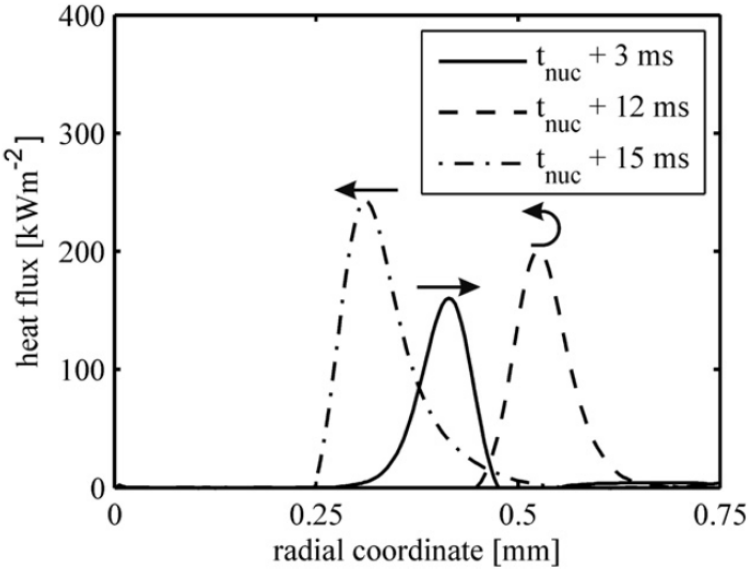

Figure 11: Local heat flux at different times during bubble growth; the arrows indicate the direction of the moving contact line (figure adapted from [48]).

Kunkelmann et al. [51] developed an experimental and numerical investigation on the local heat transfer in the contact line region during nucleate boiling and meniscus evaporation at microgravity. The working fluid was pure N-Perfluorohexane with a 0.2 K subcooled at 600 mbars. The model of Stephan and Busse [48] estimated the heat flux and evaporation rate due to microlayer effects. The authors found that for the receding contact line (growing contact line) the heat flux ratio q_{max}/q_{input} is independent from the contact line speed. However, during the advancing contact line (bubble detachment process) the heat flux ratio increases linearly with the contact line speed due to the transport of colder liquid towards the wall. The authors concluded that the high heat transfer near the contact line during

bubble growth is due to microlayer evaporation while during bubble detachment it is due to microlayer evaporation and transient conduction.

1.3 Objectives of the Present Work

The present work attempts to develop a numerical simulation of nucleate boiling with the aim of generating fundamental understanding of the boiling processes. The following lines enumerate the specific objectives of the present work:

1. Generate fundamental understanding of interfacial heat and mass transfer in phase-change through theoretical analyses of phase-change such as planar interface evaporation and spherical bubble growth.

2. Identify the requirements and propose numerical methods to simulate boiling processes. Theoretical cases of phase-change will be used as a reference to quantify the accuracy of the adopted numerical methods. In general, the simulation should have the following features:

    - Compute mass transfer with interfacial temperature gradients along the interface normal direction to generate proper bubble growth rates.
    - Include the condition of interface at the saturation temperature in the simulation in order to generate correct temperatures near the interface.
    - Simulate a sharp interface (interface within one cell) to generate correct velocity jumps and heat transfer near the interface.

3. Identify a model to predict the contribution of microlayer evaporation at the contact line to the overall bubble growth rate. The simulation of nucleate boiling should include the microlayer model to generate proper bubble growth rates.

4. Customize the software ANSYS-Fluent to perform the simulation. The numerical software solves the governing equations for momentum and energy transport, but lacks

models to account for the interfacial heat and mass transport processes.

5. Simulate nucleate boiling and compare with experimental data to validate the numerical model.

6. Use the numerical model to determine the variation of the heat transfer coefficient and wall shear stress. The heat transfer coefficient will indicate the ability of the nucleating bubble to dissipate heat, whereas the wall shear stress will show the length of influence of the nucleating bubble.

2 Proposed Work

This chapter provides details of the contribution of the present work. The contribution of the present work consists on proposing methods to simulate sharp interfaces with heat and mass transfer in nucleate boiling. In particular, the present work proposes models to: (i) compute mass transfer with interfacial temperature gradients, (ii) include a sharp interface with the saturation temperature in the simulation, and (iii) preserve the interface sharpness in the Volume-of-Fluid (VOF) interface tracking algorithm while modeling evaporation at the interface.

2.1 Proposed Mass Transfer Models

In boiling, the mass transfer across the interface depends on the interfacial temperature gradients (see section 1.2.1). An accurate estimation of the interfacial gradients is a non-trivial task since the gradients require the estimation of temperature at reference points located along a direction normal to the interface. In most interface-cells (cells with an interface), the location of the reference points differs from the centers or nodes of the computational cells (the location where the simulation finds the temperature).

The present work proposes three different methods to compute mass transfer: (i) interface-boundary method, (ii) one-cell method, and (iii) external-software method. The interface-boundary method estimates the temperature gradients by using the distance from the interface to the neighboring-cell and it is valid for horizontal or vertical interfaces. The one-cell method considers the effect of the interface location and orientation and computes the interfacial temperature gradient with a single computational cell; however, the one-cell method shows small oscillations on the magnitude of the temperature gradients along the interface and works only for first order approximations. The external-software method uses an external mathematical software (such as MATLAB) to interpolate the temperature of the reference points located in the interface normal direction. The external-software method

allows a precise estimation of the mass transfer with first and second order approximations, which is needed to detect the variation of the mass transfer along the interface. The following subsections describe the characteristics of each proposed method.

2.1.1 Interface-boundary Method

The proposed interface-boundary method allows an easy computation of the evaporative mass flux when the interface is horizontal or vertical. Therefore, the interface-boundary method is suitable for simulations of planar interface evaporation. Although the method works only for simplified scenarios, it is extremely useful to show the basic requirements of an accurate computation of the evaporative mass flux.

Consider the scenario shown in Figure 12, which consists of computational cells with a planar interface. Cells A and B are vapor-cells, cells D and E are liquid cells, and cell C is an interface-cell. The center of the computational cell A has a temperature T_A and a similar condition applies to the other cells. The interface has the saturation temperature T_{sat}. The distance from the interface to the center of the computational cell B is given by d_B, and the distance from the interface to the center of the computational cell D is given by d_D.

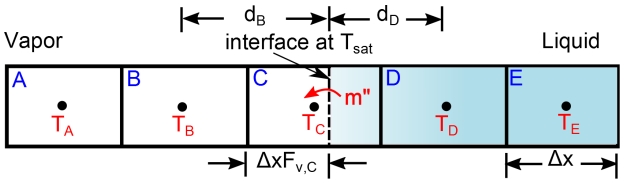

Figure 12: Proposed interface-boundary method to compute mass transfer in evaporation at a planar interface.

In the analyzed case, Eq. (4) gives the evaporative mass flux at the interface:

$$m_e'' = \frac{1}{h_{fg}} \left(-k_v \frac{dT}{dx}_{int,v} + k_l \frac{dT}{dx}_{int,l} \right), \tag{4}$$

where h_{fg} is the latent heat of evaporation, k_v and k_l are the vapor and liquid thermal conductivities, $\frac{dT}{dx}_{int,v}$ is the x-component temperature gradient at the interface on the vapor

side, and $\frac{dT}{dx}_{int,l}$ is the x-component temperature gradient at the interface on the liquid side.

Based on the conditions described in Figure 12, Eq. (5) gives the x-component temperature gradient at the interface on the vapor side and Eq. (6) gives the x-component temperature gradient at the interface on the liquid side:

$$\frac{dT}{dx}_{int,v} = (T_{sat} - T_B)/d_B \tag{5}$$

$$\frac{dT}{dx}_{int,l} = (T_D - T_{sat})/d_D. \tag{6}$$

Eqs. (5) and (6) assume a first order backward and forward approximation, respectively.

Volume-of-fluid interface tracking method uses F_v to track the interface, where F_v represent the vapor volume-fraction at the interface-cell. Eqs. (7) and (8) give the distance d_B and d_D in terms of the grid cell size ΔS and F_v, the equations assumes that the interface is horizontal:

$$d_B = 0.5\Delta s + F_v \Delta S \tag{7}$$

$$d_D = 0.5\Delta s + (1 - F_v)\Delta S. \tag{8}$$

A combination of Eqs. (4)-(8) express the evaporative mass flux at the interface in terms of the vapor volume-fraction at the interface-cell and the grid cell size given by:

$$m_e{}'' = \frac{1}{h_{fg}} \left(-k_v \frac{(T_{sat} - T_B)}{0.5\,(\Delta s + F_v)} + k_l \frac{(T_D - T_{sat})}{\Delta s\,(1.5 - F_v)} \right). \tag{9}$$

Eq. (9) assumes that the interface is horizontal and uses the interface as a boundary to compute the interfacial temperature gradients. The method applies to simulations of phase-change with planar interfaces. Eq. (9) indicates that the computation of evaporative mass flux depends on the interface location and the temperature of the neighboring-cells.

2.1.2 One-cell Method

Bubble growth in boiling consist of an interface that continuously changes its orientation. The change in interface orientation affects the direction of the normal vector, which influences the computation of the heat flux that arrives to the interface. Therefore, the mass transfer model should consider the interface orientation and its dependence with the temperature gradients at the interface along the normal direction. This section describes a proposed one-cell method to compute mass transfer at interfaces that have a non-vertical orientation. The proposed one-cell method uses a single cell to compute the temperature gradients at the interface. The advantages of the proposed method are better accuracy and simpler computation relative to available methods in the literature described in section 1.2.1.

Figure 13 illustrates the procedures used by the proposed one-cell method to determine the temperature gradient at the interface. Three probes that are perpendicular (or normal) to the interface are injected on the liquid phase. The objective of probe-1 is to locate the closest point on the interface from the interface-cell center, the objective of probe-2 is to identify the cell that is closest to the interface in the normal direction (the G-cell, where G comes from gradient). Probe-1 and probe-2 support the construction of probe-3. Probe-3 is special due to its orientation and its length; this probe is normal to the interface, and has a length such that the tip of the probe intersects a cell center on the liquid. After the injection of the three probes, mathematical expressions find the mass transfer through the interface and the fixed temperature at the interface-cells.

It is important to emphasize that most of the previous methods in the technical literature use probe-1 to identify the interface. Udaykumar and Shyy [2] used probe-2 to estimate the temperature gradient at a reference point, see Figure 2(a). The method of Sato and Niceno [3] is independent of probe-1 and probe-2, but it requires horizontal or vertical probes to identify two points at the interface, see Figure 2(b). Therefore, the proposed method adds only one extra probe (probe-3), which is defined based on probe-2 and on the vector normal to the interface. The steps to inject the probes are described in the following lines:

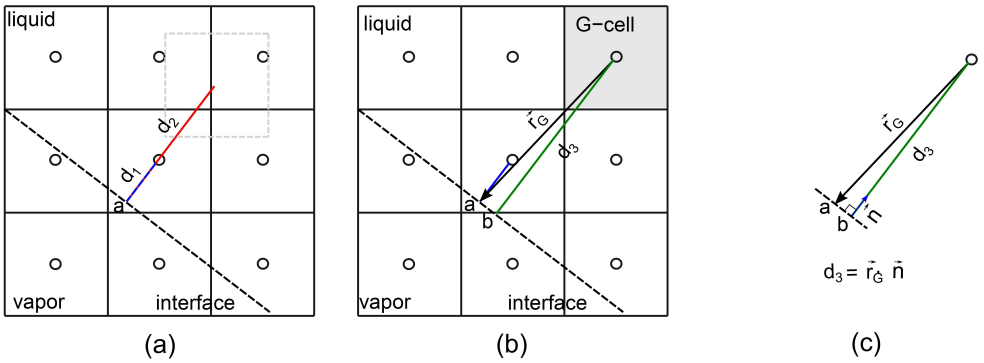

Figure 13: Proposed once-cell algorithm to evaluate the temperature gradient at the interface. (a) Steps 1) to 4), (b) Steps 5) to 7), (c) alignment of G-cell center with normal vector.

1. Identify the interface-cell where the mass transfer and the fixed temperature value should be determined.

2. Inject probe-1. The origin of this probe is the interface-cell center, the orientation is normal to the interface, and the length is such that the tip intersects the interface. The point that intersects the probe with the interface is called "point a".

3. Inject probe-2. The origin of this probe is "point a", the orientation is normal to the interface, and the length is d_2.

4. Explore a region around the tip of probe-2 to find the center of the cell that is nearest to the interface in the normal direction. The name of the nearest cell is "G-cell".

5. Inject probe-3. The origin of this probe is the cell center of the G-cell, the orientation is normal to the interface, and the length is determined with steps 6) and 7).

6. Create a vector $\overrightarrow{r_G}$ connecting the center of the G-cell with the "point a" at the interface.

7. Determine the length of probe-3 by the projection of the vector $\overrightarrow{r_G}$ on the normal vector $\hat{n}$. The name of the intersection between probe-3 and the interface is "point b".

In the procedure of injecting probes, the length of probe-2 d_2 (in step 3) is such that the tip of probe-2 lies outside of the interface-cell. A proper length of probe-2 could be determined based on the size of the interface-cell. The tip of the probe lies outside the interface-cell if the length is larger than the maximum distance between two points on the periphery of the interface-cell. This distance is given by $\sqrt{\Delta x^2 + \Delta y^2}$, where Δx and Δy are the length of the cell in the x and y directions, respectively.

Another important factor is the region explored to find the nearest cell center in step 4. This step ensures that only the nearest cell center is inside the perimeter of exploration. To perform this task, the center of the explored region should be the tip of the probe-2, and the length of the explored region could be Δx in the x-direction and Δy in the y-direction.

In step 7, Eq. (10) determines the length of probe-3:

$$d_3 = \overrightarrow{r_G} \cdot \hat{n}, \tag{10}$$

where $\hat{n}$ is the unit vector that is normal to the interface. It is important to observe that the proposed method aligns the center of the G-cell with the normal vector. The projection of the vector $\overrightarrow{r_G}$ on the normal vector gives the distance (d_3) from the center of the G-cell to the interface along the normal direction, see Figure 13(c). Therefore, it is valid to use d_3 to determine the temperature gradient. The only drawback is that the gradient ignores the center of the interface ("point a" in Figure 13(b)). Instead, the gradient considers a point near the center ("point b" in Figure 13(b)). In other words, the method translates the normal vector along the interface to align the normal gradient with the center of the G-cell. The translation assumes that the interface has a linear shape. Such a translation might introduce inaccuracies, which could become significant with highly deformed interfaces.

Another important fact is that the proposed approach requires structured meshes near the interface. This is a requirement to identify the G-cell. The simulation in the present work uses an interface reconstruction algorithm to define the interface. In addition, the

present simulation considers the interface location to fix the temperature of the interface-cells. Under these conditions, the use of structure meshes is acceptable. Previous works have used unstructured grids with moving meshes to increase numerical accuracy by augmenting grid resolution near the interface [52,53]. In the case of simulations with unstructured meshes, the proposed method should be modified to account for the nonorthogonal cell shape.

Particularly, the one-cell method estimates the normal temperature gradient at "point b" (see Figure 13). The normal gradient at point-b is given by the difference of the temperatures at extremes of the probes divided by the length of the probes as given by:

$$\frac{\partial T}{\partial n}_{b} = \frac{T_G - T_{sat}}{d_3}. \tag{11}$$

Therefore, the proposed one-cell method estimates the mass transfer at the interface as:

$$m_e'' = \frac{k_l}{h_{fg}} \frac{\partial T}{\partial n}_{b} = \frac{k_l}{h_{fg}} \frac{T_G - T_{sat}}{d_3}. \tag{12}$$

The one-cell method requires the normal vector to identify the G-cell and to determine the distance d_3. The main advantage is that the method estimates the temperature gradient by using the temperature of only one computational cell (previous method require the normal vector and the identification of multiple cells). In addition, the proposed one-cell method provides a direct estimation of the temperature gradient whereas the method that uses an interpolation function requires multiple operations to determine the coefficient of the interpolation function. However, one disadvantage of the one-cell model is that the distance d_3 changes at each interface-cell based on the interface orientation and position. The change in the distance d_3 leads to small oscillations of the temperature gradients that could affect the accuracy of the simulation. In addition, the one-cell method obtains temperature gradients only with first order approximations.

2.1.3 External-software Method

The present section describes a proposed external-software method that uses an external software (such as MATLAB) to compute the temperature gradients in a direction normal to the interface. The external-software method allows the estimation of the temperature gradients with first and second order approximations.

The proposed external-software method uses MATLAB to interpolate the temperature of reference points in the estimation of the interfacial gradients, see Figure 14(b). MATLAB has functions that perform interpolations within a set of known data points. Therefore, MATLAB can use data of the temperature field near the interface and the coordinates of the reference points to perform the interpolation. The procedure consists on the following steps (see Figure 14(c)):

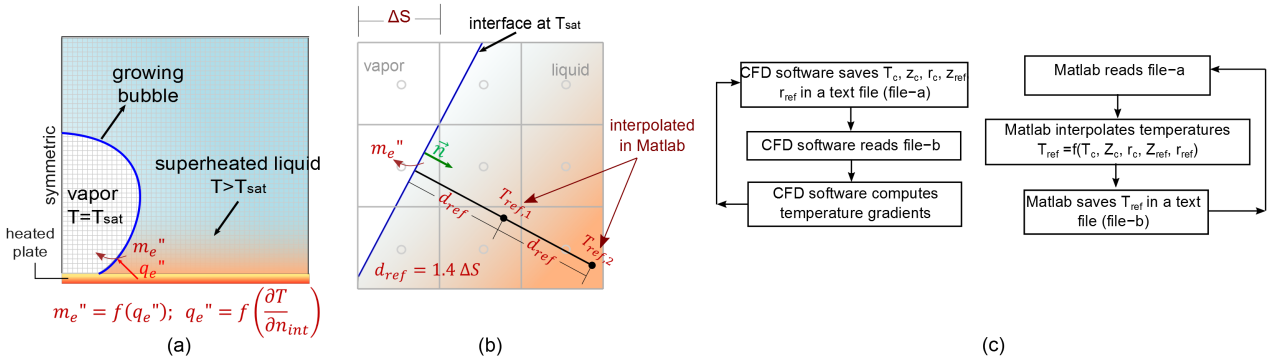

Figure 14: Proposed method to compute temperature gradients with an external software: (a) diagram of nucleating bubble, (b) proposed external-software method, (c) diagram of method to communicate CFD-software and MATLAB.

1. The CFD software writes a file (file-a) with the temperature and coordinates of the cells near the interface and the coordinates of the reference points.

2. MATLAB reads the file-a and uses interpolation functions to estimate the temperature at the reference points.

3. MATLAB writes a file (file-b) with the temperature of the reference points.

4. The CFD software reads the file-b and computes the interfacial temperature gradients.

During the simulation, the CFD software and MATLAB run simultaneously but not necessarily synchronized. The method can execute steps (2) and (3) multiple times before the CFD software writes or reads a file without affecting the numerical solution. We used the function *movefile* to ensure that MATLAB writes only one file (file-b) instead of multiple files. In MATLAB, the function $F = scatteredInterpolant(r_c, z_c, T_c)$ was used to create the field of temperatures near the interface by using the location of the center of the computational cells r_c, z_c and the temperature at the center of the cells T_c. The function $F(r_{ref}, z_{ref})$ interpolated the temperature at the reference point.

The present work takes advantage of the proposed method to analyze the effect of assuming first and second order approximations in the estimation of the interfacial temperature gradients. To our knowledge, there is no report in the technical literature that evaluates the effect of the order of approximation on the estimation of mass transfer in multiphase simulations. However, such study might be useful to optimize the performance of the numerical model.

Eqs. (13) and (14) show the approximations used to estimate the interfacial temperature gradients. Eqs. (13) and (14) corresponds to a first order and second order backward approximation, respectively:

$$\frac{\partial T}{\partial n}_{int,1st} = \frac{T_{ref,1} - T_{sat}}{d_{ref}} \tag{13}$$

$$\frac{\partial T}{\partial n}_{int,2nd} = \frac{-T_{ref,2} + 4T_{ref,1} - 3T_{sat}}{2d_{ref}}, \tag{14}$$

where the reference distance d_{ref} was defined based on the grid cell size ΔS as $d_{ref} = \sqrt{2}\Delta S$. The factor of $\sqrt{2}$ ensures that the reference point lies outside the interface-cell, which contributes to a proper interpolation.

The proposed external-software method requires writing and reading files and the use of an external mathematical software. However, the method allows an easy and accurate interpolation of the temperature at multiple reference points, which improves the overall

accuracy of the simulation. MATLAB performs the interpolation by creating a complete map of the temperature field near the interface instead of looking into specific cells around the reference points. In addition, the method performs the interpolation without subroutines to estimates the coefficients of the interpolation function. Another advantage is that the method is extremely accurate since the interpolation functions consider advanced algorithms. The proposed method applies to boiling that depends on interfacial temperature gradients or droplet generation in microreactors that depends on interfacial species concentration gradients [54]. In addition, a similar approach could perform simulations that require advanced functions to change the conditions in the simulation in real-time.

2.2 Proposed Models for Interface Temperature

In the simulation of nucleate boiling, the interface is an internal and dynamic boundary with the saturation temperature that travels across computational cells. Various approaches impose the interface condition in simulations of boiling (see section 1.2.2): (i) assume a transition region with effective conductivies, (ii) modify the length of the neighboring-cells, and (iii) interpolate the temperature of the interface-cells (ghost fluid method).

The present work proposes two different methods to include the interface saturation temperature in the simulation: (i) interface-boundary method, and (ii) interpolation of interface-cell temperature with a linear temperature profile approximation. The interface-boundary method modifies the discretization schemes of the neighboring-cells to include the saturation temperature of the interface. The modified discretization schemes consider the interface as a boundary. The interpolation of interface-cells fixes the temperature of the interface-cells with an interpolated temperature. The interpolated temperature considers the interface temperature and location. The following subsections describe the characteristics of each proposed method.

2.2.1 Interface-boundary Method

The proposed interface-boundary method is valid only for horizontal or vertical interfaces. However, the method provides the basic principles to perform accurate simulations of multi-phase flows where the interface has a constant temperature. The interface-boundary method modifies the discretization schemes at the neighboring-cells to include the interface saturation temperature. The modification uses the interface as a boundary to perform the integration of the terms in the energy equation. To illustrate the method, consider the situation depicted in Figure 15 with a moving interface at saturation temperature embedded into the control volumes with centers at W, P, E, and EE. The cell with center at W has a face on the west side at ww, whereas the cell with center at E has a face on the east side at ee. This is done to have a complete definition of the control volumes around an interface. The cell with center E is known as the interface-cell because it has an interface within it. The cells with centers at P and EE are known as the neighboring-cells because they are next to the interface-cell. The objective is to include the interface effect in the discretization of neighboring-cell P (similar procedures apply to the discretization of the neighboring-cell EE).

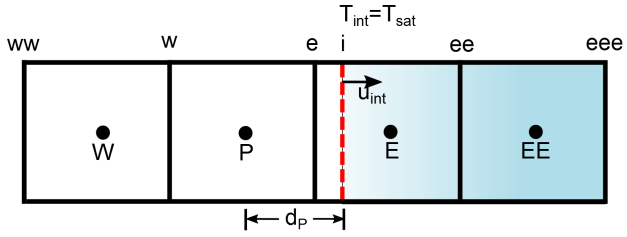

Figure 15: Proposed interface boundary method to include the interface temperature in simulations of planar interface evaporation.

Conventionally, the integration of the terms in the heat conduction equation is done from face to face at each cell. In other words, the integration is performed from ww to w, from w to e, and from e to ee. However, this procedure is no longer applicable since it ignores the effect of the interface in the solution of the energy equation. This method gives inaccurate temperatures at the interface cell and neighboring-cells, which is a serious problem because the mass transfer (as discussed in section 2.1) depends on the temperature gradients at the

interface. Therefore, ignoring interface effects in the solution of the energy equation leads to erroneous interface displacements.

A first attempt to solve this problem could be to decrease the grid cell size. It may be thought that with a smaller grid the errors of ignoring the interface position should become smaller. However, smaller grids may result in higher errors since more cells ignore the interface in a given time interval.

To solve this problem, it is necessary to take into account the interface effects. The proposed interface-boundary method integrates from face to interface instead of integrating from face to face. The method estimates the gradients at the faces with central differences and the gradients at the interface with first order backward or forward approximations. The integration at the neighboring cell with center P is performed from face w to i (where i is the interface position), and the integration in the neighboring cell with center EE is done from i to eee. In this procedure, P and EE are set at a fixed position. This implies that with the proposed method, the grid remains unchanged, and only the way that the fluxes are computed is modified.

Eq. 15 gives the integral of the diffusive terms in heat conduction equation over the neighboring cell from w to i (a similar equation gives the integral on the liquid side from i to eee):

$$\int_w^i k \frac{\partial}{\partial x}\left(\frac{\partial T}{\partial x}\right) dx = \left[\left(k\frac{\partial T}{\partial x}\right)_i - \left(k\frac{\partial T}{\partial x}\right)_w\right], \tag{15}$$

where, the temperature gradient at w is obtained with second order central differences, and the temperature gradient at i is obtained with first order backward differences. Then, Eq. (15) becomes:

$$\int_w^i k \frac{\partial}{\partial x}\left(\frac{\partial T}{\partial x}\right) dx = \frac{k_i\left(T_i - T_P\right)}{d_p} - \frac{k_w\left(T_P - T_W\right)}{\Delta x}, \tag{16}$$

where d_P is the distance from the interface to P (see Figure 15), k_i in the thermal conductivity at i, k_w is the thermal conductivity at w, Δx_v is the distance of integration from w to i. The

main advantage of using backward differences for the heat flux at the interface is that k_i can be expressed as k_v, where k_v is the thermal conductivity of the vapor, which implies that k_i is not considered as an average conductivity. Also, a backward difference approximation takes into account the interface temperature, T_{sat}. However, the main disadvantage of backward differences is that they are only first-order accurate. Therefore, this approach sacrifices accuracy in the estimation of the temperatures of the neighboring-cells. On the other hand, the approach reduces the errors significantly by taking into account the interface position and temperature.

The Volume-of-Fluid (VOF) method uses the variable F_v to track the interface. F_v represents the vapor volume-fraction withing the cell. Eq. (17) use the vapor volume-fractions to determine the distance d_P used to estimate the temperature gradient at the interface:

$$d_P = 0.5\Delta x + F_v \Delta x. \tag{17}$$

The proposed interface boundary method uses Eq. (16) to include the effect of the interface in the diffusive terms in the energy equation. A similar procedure should be applied to discretize the convective terms if the fluid moves with a certain velocity. The method uses the interface as a boundary in the discretization of the neighboring-cells and avoids the definition of average properties at the interface-cells. However, the method requires the modification of the discretization schemes on the neighboring-cells, which is a nontrivial task in 2D and 3D simulations. In numerical packages, the method should eliminate the traditional discretization schemes and apply the modified discretization schemes. In addition, the method is unable to define a distance d_P in cases where the interface-cells have a small interface oriented at a certain angle. In such cases, the method should interpolate the temperature of the interface-cells. Sato and Niceno [3] developed a similar method to include the interface conditions in 2D and 3D simulations, but the authors estimate the temperature gradients at the neighboring cell centers based on the interface position.

2.2.2 Linear Temperature Profile Approximation

Simulations of boiling have an interface that continuously changes its orientation. In addition, the interface has the saturation temperature that should affect the temperature of the neighboring-cells (see Figure 16(a)). The modeling of the interface could be done by modifying the discretization schemes of the neighboring-cells to include the interface condition as described in the previous section. However, the modification of the discretization schemes becomes more challenging in 2D and 3D simulations specially when using a numerical package to perform the simulation (the modification should eliminate the conventional procedures and apply the modified procedures).

A more effective approach assumes that the interface-cells have fluid properties and interpolates the temperature of the interface-cells by using interface temperature and location (ghost-fluid method, see section 1.2.2). The advantage of the ghost-fluid method is that it considers conventional discretization schemes on the neighboring-cells (the neighboring-cells read the interface location and temperature by taking into account the interpolated interface-temperature in the discretization of the energy equation). However, the method requires an interpolation function (refer to section 1.2.2 for a discussion of the various methods available to interpolate the interface-cell temperature).

The present work proposes a method to interpolate the temperature of the interface-cells in the ghost-fluid method. The proposed method interpolates the temperature of the interface-cells with a linear temperature profile approximation in a direction normal to the interface (see Figure 16(b) and (c)). The proposed method uses the interfacial tempeature gradients to define the slope of the linear temperature profile. Based on these assumptions, Eq. (18) provides the equation to estimate the temperature of the interface-cells T_{IC}:

$$T_{IC} = T_{sat} + \left(\frac{\partial T}{\partial n}_{int} \right) d_{IC}, \tag{18}$$

where T_{sat} is the interface temperature, $\frac{\partial T}{\partial n}_{int}$ is the interfacial temperature gradient along

the normal direction, and d_{IC} is the distance from the interface to the interface-cell (IC) center along the normal direction (see Figure 16(c)).

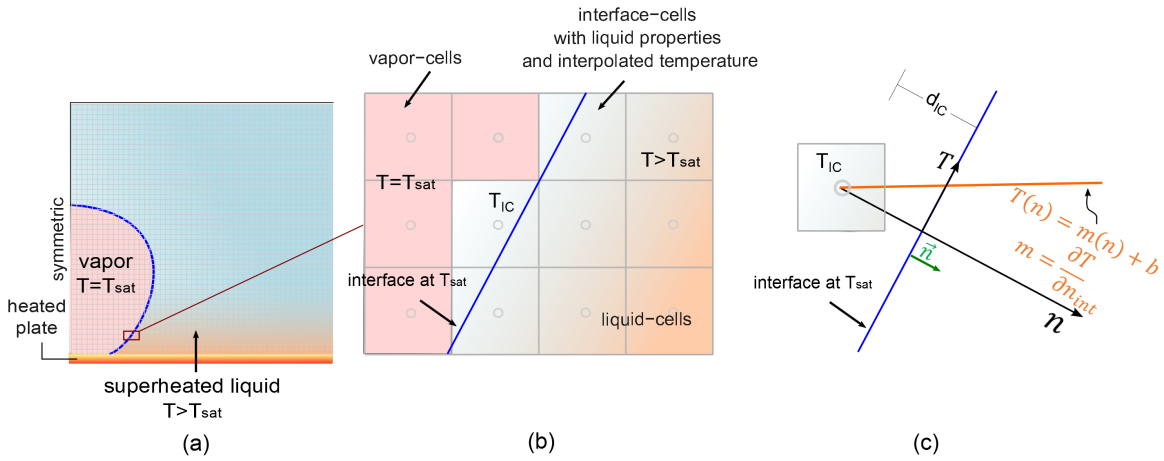

Figure 16: Interpolation of interface-cell temperature:(a) diagram of nucleating bubble, (b) ghost fluid method, (c) proposed linear temperature profile approximation along normal direction.

The proposed method requires the magnitude of the temperature gradient at the interface along the normal direction. Such a magnitude can be obtained with the proposed one-cell method or external-software method (discussed in section 2.1).

One advantage of the proposed method is that it provides a direct estimation of the interfacial temperature rather than solving partial differential equations as done in previous works [31, 32]. Another advantage of the proposed method lies in its accuracy (see section 5.2.4). The linear temperature profile is a common and accurate approximation used in numerical methods to determine the heat fluxes or temperature gradients at the faces of the computational cells. Such temperature gradients participate in the discretization of the energy equation at each computational cell. In fact, the conventional estimation of temperature gradients at the faces of the cells with first order approximations (backward or forward differences) or second order approximations (central differences) comes from the use of Taylor series that consider the terms until the first derivative. The consideration of only the first derivative implies that the methods assume a linear temperature profile approximation. Versteeg and Malalasekera [55] proved that the use of a linear profile assumption to calculate

the gradients at the faces of the computational cells give accurate results even for coarse grids. In addition, the authors showed that other higher-order approximations (e.g. QUICK) give the same accuracy with larger grid cell sizes. In the present work, a grid-independence analysis is done to ensure that the numerical results are independent of the grid cell size and that the simulations use an optimal grid (see section 5.4.2 and 5.5.2).

Special numerical methods fix the interpolated temperature at the interface-cells. The numerical methods should provide the interpolated values as an output of the numerical solution to ensure that the interpolated temperature is part of the discretized equations of the neighboring-cells. In other words, the neighboring-cells should see the interface as an internal boundary. To perform this task, we adopted the large coefficient method proposed by Patankar [56]. Sun et al. [57] used the large coefficients method to fix the saturation temperature of the interface-cells in ANSYS-Fluent in the simulation of film boiling; however, the interpolation function used temperature gradient at cell centers rather than at the interface. To illustrate the method, consider that the discretization equation of the energy equation on the computational cell "P" is given by [58]:

$$a_P T_P = a_W T_W + a_E T_E + a_S T_S + a_N T_N + S, \tag{19}$$

where a and T represent the coefficients and temperature of the cells P, W, E, S, and N, respectively (W, E, S, and N stand for west, east, south, and north sides of the cell P). In addition, S is a source term. The large coefficient method uses Eq. (20) to define the source term S at the interface-cells:

$$S = T_{IC} C - T_P C, \tag{20}$$

where C is a large coefficient, T_{IC} is given by Eq. (18), and T_P is the temperature of the cell P given by the numerical model. Since the coefficient C is large, the source term becomes much greater than the rest of the terms in Eq. (19). Therefore, the substitution of Eq. (20)

in Eq. (19) gives:

$$T_P = T_{IC}. \tag{21}$$

The large coefficient method fixes the interpolated temperature T_{IC} at the computational cell P. In addition, the neighboring-cells see the interface since the discretization equation on these cells use the interpolated value. The method eliminates the use of average properties on the interface-cells. At each iteration, the simulation declares the source term given by Eq. (20) at the interface-cells. The temperature of the neighboring-cells changes based on the source term. Therefore, in the next iteration the source term changes based on the new temperature of the neighboring-cells. The procedure repeats until the temperature of the neighboring-cells and the source term achieve a constant value (convergence criteria is satisfied). Usually, convergence occurs in less than five iterations.

The proposed method estimates the temperature of the interface-cells to include the condition of a sharp interface with the saturation temperature in the simulation. The proposed sharp interface method provides a direct estimation of the interpolation temperature rather than solving additional differential equations. The assumption of a linear temperature profile leads to an accurate estimation of the interface-cell temperature since the distance between the interface-cell center and the interface is small (of the order of the grid cell size). In addition, the method fixes the temperature of the interface-cells to impose the interface condition. The large coefficients method avoids the modification of discretization schemes on neighboring-cells, which makes it compatible with CFD packages or developed in-house codes.

2.3 Proposed Model to Preserve Interface Sharpness

Numerical simulations of multiphase flows employ algorithms to define and track a sharp interface [59] (see section 1.2.3). Algorithms that use volumes to track the interface such as Volume-of-Fluid (VOF) employ additional techniques to advect fluid through the cell faces

(e.g. donor-acceptor, direction-split, and a piecewise-linear interface) in a way that prevents the interface smearing [39, 42, 60–62]. In addition, boiling simulations with VOF should use techniques to define mass transfer in a way that is compatible with the interface-tracking algorithm. Available techniques to keep the interface sharp while declaring mass transfer include interface smearing into a restricted number of cells with an interface-sharpening equation [3,25–27], the distribution of mass source terms in multiple cells around the interface [9–11,13–15], and the advection of the interfacial mass flux through the faces of the interface-cells with a split approach [40, 63–66].

The present work describes an interface-cells segregation algorithm to preserve the interface sharpness in the simulation of nucleate boiling with VOF sharp interface method. To illustrate the segregation method, consider computational cells and the interface shown in Figure 17, where A-G are interface-cells. The proposed segregation algorithm identifies cells A, C, E, and G as "mass-transfer-cells" and cells B, D and F as "unsuitable-interface-cells". The method declares mass transfer only at mass-transfer-cells and declares zero mass transfer at unsuitable-interface-cells. To avoid the loss of mass, the interface area of the unsuitable-interface-cells adds to the area of its neighboring-mass-transfer-cell. As the bubble grows, cells A, C, E, and G become vapor-cells and cells B, D, and F become mass-transfer-cells. The proposed segregation algorithm keeps the interface sharp by declaring mass transfer only at the interface-cells that lie next to vapor-cells to gain vapor. The interface remains sharp because interface-cells that lie far (one cell away) from vapor-cells have zero mass transfer, which prevents the smearing of the interface.

Various ways can classify the interface-cells as mass-transfer-cells and unsuitable-interface-cells. Here we describe an algorithm that is simple and that uses volume-fractions to perform the classification. To identify the mass-transfer-cells, the simulation loops over all the faces (edges) of the computational cells. While visiting the faces, the simulation retrieves the volume-fraction of the two adjacent cells that share the same face. A cell is a mass-transfer-cell if its liquid volume-fraction is higher than zero and if it lies next to a vapor cell (liquid

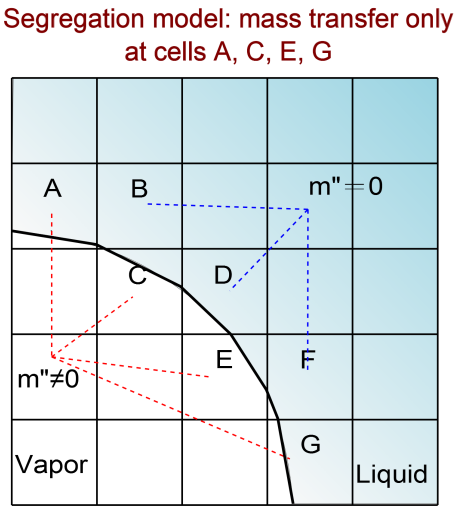

Figure 17: Proposed model to keep the interface sharp in the simulation.

volume-fraction equal to zero). A similar procedure identifies unsuitable-interface-cells: a cell is an unsuitable-interface-cell if its liquid-volume of fraction is between zero and one and if it lies next to a mass-transfer-cell.

The proposed segregation method allows the simulation of nucleate boiling with the VOF sharp interface model. The segregation method declares mass transfer only at the interface-cells, which contributes to achieve accurate velocity jumps at the interface. In addition, the method eliminates the need to use differential equations to distribute mass source terms or to keep the interface sharpness, which reduces the complexity of the simulation.

3 Numerical Model

This section is divided into three subsections. Subsection 3.1 shows the governing equations for mass, momentum, energy, and interface-tracking. Subsection 3.2 focuses on the model used to account for heat and mass transfer at the contact line. Subsection 3.3 describes the customization of ANSYS-Fluent developed to perform the simulation.

3.1 Governing Equations

The Volume-of-Fluid (VOF) equation that tracks the interface is given by:

$$\frac{\partial F_v}{\partial t} + \nabla \cdot (F_v \vec{v_v}) = \frac{{m_e}''}{\rho_v} \frac{A_i}{V_c}, \tag{22}$$

where F_v is the vapor volume-fraction on each cell, ρ_v is the vapor density, $\vec{v_v}$ is the vapor velocity, ${m_e}''$ is the evaporative mass flux at each interface-cell, A_i is the interface surface area within each interface-cell, and V_c the cell volume. In the computational domain, cells filled with vapor have $F_v = 1$, cells filled with liquid have $F_v = 0$, and cells that have an interface have $0 < F_v < 1$. Eq. (23) is a constraint for the liquid volume fraction, F_l, at each cell:

$$F_l + F_v = 1. \tag{23}$$

The piecewise-linear scheme reconstructs the interface with the assumption that the interface shape in each cell is a straight line. The interface inclination and the volume-fraction of each cell determine the interface surface area A_i [67]. Eq. (24) estimates the evaporative mass flux at each interface-cell:

$${m_e}'' = \frac{k_l}{h_{fg}} \frac{\partial T}{\partial n}_{int}, \tag{24}$$

where k_l is the liquid thermal conductivity, h_{fg} is the latent heat of evaporation, and $\frac{\partial T}{\partial n}_{int}$ is the temperature gradient at the interface along the normal direction. The main challenge

in Eq. (24) is the estimation of the temperature gradient at the interface. The interfacial temperature gradient requires a temperature reconstruction algorithm. As previously stated, the present work proposes models to estimate the normal interfacial temperature gradients (see section 2.1). The proposed models are easy to implement and improve the accuracy of the mass transfer estimation.

The momentum conservation equation that finds the pressure and the velocity on the liquid and vapor phases is given by:

$$\rho\left[\frac{\partial\vec{v}}{\partial t}+\left(\nabla\cdot\vec{v}\right)\vec{v}\right]=-\nabla p+\nabla\cdot\mu\nabla\vec{v}+\nabla\cdot\mu\nabla\vec{v}^{T}+\rho\vec{g}+S_{M}, \tag{25}$$

where p is the pressure, μ is the viscosity, g is the gravitational constant, and S_M accounts for the surface tension effects. The continuous surface force (CSF) model proposed by Brackbill et al. [68] estimates the surface tension volume force:

$$S_{M}=\sigma\frac{\rho K\nabla F_{v}}{0.5\left(\rho_{v}+\rho_{l}\right)}, \tag{26}$$

where σ is the surface tension. The interface curvature, K, is estimated as:

$$K=\nabla\cdot\hat{n}, \tag{27}$$

where the unit vector normal to the interface is given by:

$$\hat{n}=\frac{\nabla F_{v}}{|\nabla F_{v}|}. \tag{28}$$

The simulation uses the Green-Gauss theorem to compute the gradients of any scalar at the cell center based on the volume-fractions at the faces. The present work explores three methods to estimate the volume-fraction at the faces. The node-based gradients method: finds the volume-fractions at the faces by the arithmetic average of the volume-fractions at

the nodes. The cell-based gradients method: finds the volume-fractions at the faces by the arithmetic average of the volume-fractions at the neighboring cell centers. The node-based gradients method with smoothed volume-fractions: finds the volume-fractions at the faces with smoothed volume-fractions at the nodes. ANSYS-Fluent provides options to smooth the volume fractions in the estimation of surface tension. For details on the estimation of the gradients, the reader is directed to ANSYS-Fluent documentation.

The energy conservation equation that provides the temperature for both liquid and vapor phases is given by:

$$\frac{\partial T}{\partial t} + \nabla \cdot \vec{v} T = \nabla \cdot \alpha \nabla T + S_T, \tag{29}$$

where α is the thermal diffusivity, and S_T is a source term. The source term fixes the temperature of the interface-cells to properly account for the sharp interface. Similar to other simulations with a sharp interface [3, 7], no source term was added at the interface-cells to account for latent heat of evaporation. This is because the simulation considers the interface as a boundary condition by modifying the discretization schemes of the neighboring-cells or by interpolating the temperature of the interface-cells (see section 2.2). Latent heat effects should be considered only at the interface and not on the liquid or vapor phases. In the simulation, the temperature of the surrounding liquid is therefore dictated by the interface temperature and position.

In Eqs. (25), (26), and (29) the density, viscosity, and thermal diffusivity are determined with the vapor volume fraction:

$$\vartheta = F_v \vartheta_v + (1 - F_v)\, \vartheta_l, \tag{30}$$

where ϑ represents the density, viscosity, or thermal diffusivity. It is important to highlight that in the present work, the energy equation considers a sharp interface by using the proposed methods described in section 2.2. Therefore, the properties used to estimate the temperature of the interface-cells are the properties of the liquid, and not a combina-

tion of liquid and vapor properties. This technique increases the accuracy of the prediction of the temperature field near the interface, which is translated into more precise interface displacements.

3.2 Microlayer Model

The present simulation includes microlayer effects by considering the model proposed by Wayner et al. [45] with the modifications of Kamotani [47]. In addition, the coupling and solution of the microlayer model follows the procedures in Stephan and Busse [48] and Stephan and Hammer [50]. The microlayer model consists of a system of ordinary differential equations (ODEs) that provide the variation of the microlayer profile, the microlayer slope, the capillary pressure, and the integrated heat flux.

In the microlayer model, the momentum equation for 1D fully developed laminar flow Eq. (31) gives the dependence between the fluid velocity and the change in the liquid pressure in the microlayer:

$$\frac{dp_l}{d\xi} = \mu \frac{d^2u}{d\eta^2}, \tag{31}$$

where ξ and η are the radial and axial coordinates respectively (see Figure 9(a)), p_l is the liquid pressure, and u is the fluid velocity along the radial direction. The assumptions of non-slip condition at the liquid solid interface and non-shear stress at the liquid vapor interface give the velocity distribution on the liquid film:

$$u\left(\eta\right) = \frac{1}{\mu}\left(\frac{dp_l}{d\xi}\right)\left(\frac{\eta^2}{2} - \delta\eta\right), \tag{32}$$

where δ is the film thickness. The integral of the liquid velocity along the microlayer thickness gives the mass flux in the microlayer at the location ξ:

$$m_{mic}{}'' = \rho\bar{u} = \rho\frac{1}{\delta}\int_0^\delta u(\eta)d\eta = -\frac{\delta^2}{3\nu}\frac{dp_l}{d\xi}, \tag{33}$$

where ν is the dynamic viscosity. $\Gamma = {m_{mic}}''\delta$ gives the mass flux per unit contact line length. The change in the mass flux per unit contact line at the location ξ gives the evaporative mass flux leaving the liquid film:

$$ {m_e}'' = \frac{d\Gamma}{d\xi} = -\frac{1}{3\nu}\frac{d}{d\xi}\left(\delta^3 \frac{dp_l}{d\xi}\right), \tag{34} $$

Wayner et al. [45] used a kinetic theory model and the Clausius-Clapeyron equation to express the evaporative mass flux in terms of the temperature difference and pressure difference at the interface as given by:

$$ {m_e}'' = \left(\frac{2f}{2-f}\right)\left(\frac{M_v}{2\pi R T_{sat}}\right)^{1/2}\left[\frac{M_v p_v h_{fg}}{R T_{sat} T_{lv}}\left(T_{lv} - T_{sat}\right) + \frac{V_{m,l} p_v}{R T_{lv}}\left(p_l - p_v\right)\right], \tag{35} $$

where f is an evaporation coefficient, M_v is the vapor molecular weight, R is the universal gas constant, T_{sat} is the saturation temperature, T_{lv} is the temperature of the liquid-vapor interface, $V_{m,l}$ is the liquid molar volume, and p_v is the vapor pressure at the saturation temperature. The assumptions of an ideal gas applicable to vapor and mass conservation during evaporation $\rho_v V_v = \rho_l V_l$ transform Eq. (35) into Eq. (36):

$$ {m_e}'' = \left(\frac{2-f}{f}\right)\left(\frac{M_v}{2\pi R T_{sat}}\right)^{1/2}\frac{h_{fg}\rho_v}{T_{sat}}\left[\left(T_{lv} - T_{sat}\right) + \frac{T_{sat}}{\rho_l h_{fg}}\left(p_l - p_v\right)\right]. \tag{36} $$

Recognizing that ${q_e}'' = {m_e}'' h_{fg}$ (where h_{fg} is the latent heat of evaporation), Eq. (37) expresses the interfacial heat flux in the microlayer:

$$ {q_e}'' = \frac{1}{R_{int}}\left[\left(T_{lv} - T_{sat}\right) + \frac{T_{sat}}{\rho_l h_{fg}}\left(p_l - p_v\right)\right], \tag{37} $$

where R_{int} is the interfacial resistance defined as:

$$ R_{int} = \left(\frac{2-f}{f}\right)\frac{T_{sat}\sqrt{2\pi R_v T_{sat}}}{{h_{fg}}^2 \rho_v}. \tag{38} $$

Kamotani [47] assumed 1D heat conduction through the liquid film in a direction normal to the surface and a linear temperature distribution on the liquid film. Under these conditions, Eq. (39) gives the heat flux through the microlayer:

$$q_e'' = k_l \frac{(T_{w,mic} - T_{lv})}{\delta}, \tag{39}$$

where $T_{w,mic}$ is the surface temperature at the microlayer. The combination of Eqs. (37) and (39) gives the relation between the heat flux, the surface temperature, the saturation temperature, the pressure, and the microlayer thickness given by:

$$q_e'' = \frac{(T_{w,mic} - T_{sat}) + \frac{T_{sat}}{\rho_l h_{fg}} (p_l - p_v)}{R_{int} + \delta/k_l}. \tag{40}$$

The combination of Eq. (34), Eq. (40), and $q_e'' = m_e'' h_{fg}$ generates a fourth order differential equation to model the microlayer given by:

$$\frac{(T_{w,mic} - T_{sat}) + \frac{T_{sat}}{\rho_l h_{fg}} (p_l - p_v)}{R_{int} + \delta/k_l} = -\frac{h_{fg}}{3\nu} \frac{d}{d\xi} \left(\delta^3 \frac{dp_l}{d\xi} \right). \tag{41}$$

In the liquid film, the pressure depends on the interface curvature and the molecular forces between the liquid and the surface. Wayner [69, 70] extended the Young-Laplace equation to account for the effect of the intermolecular forces in the liquid film on the liquid pressure given by:

$$p_l - p_v = -\sigma K + p_d, \tag{42}$$

where the interface curvature K is defined as:

$$K = \frac{\frac{d^2\delta}{d\xi^2}}{\left(1 + \left(\frac{d\delta}{d\xi}\right)^2\right)^{3/2}}. \tag{43}$$

Derjaguin and Zorin [46] defined the relation between disjoining pressure p_d and micro-

layer thickness as:

$$p_d = -\frac{A}{\delta^3}, \tag{44}$$

where A is the Hamaker constant, which accounts for the effect of the molecular forces. The solution of the microlayer model gives the liquid film thickness, the variation of the liquid pressure, and the heat flux as function of the position ξ in the microlayer. A dimensionless version of the microlayer model makes the numerical solution more efficient. The procedure requires the definition of the following dimensionless parameters:

$$\xi^* = \frac{\xi}{\delta_{ad}}, \quad \delta^* = \frac{\delta}{\delta_{ad}}, \quad \delta'^* = \delta', \quad \Delta {p_c}^* = \frac{\Delta p_c}{\Delta p_{c,ad}}, \quad \dot{Q}^* = \frac{\dot{Q}_{mic}}{k_l \left(T_w - T_{sat}\right)}, \tag{45}$$

where δ' is the microlayer slope ($\delta' = \frac{d\delta}{d\xi}$), Δp_c is the capillary pressure, and $\dot{Q}_{mic}$ is the integrated heat flux. The subscript "ad" stands for adsorbed film. The integrated heat flux represents the interfacial heat at the microlayer per contact line length given by:

$$\dot{Q}_{mic} = \int_{\xi_{ad}}^{\xi_{mic}} {q_e}'' d\xi. \tag{46}$$

Eqs. (47)-(50) define the dimensionless microlayer model:

$$\frac{d\delta^*}{d\xi^*} = \delta'^* \tag{47}$$

$$\frac{d\delta'^*}{d\xi^*} = D_1 \left(1 + \delta'^{*2}\right)^{3/2} \left(\Delta p^* - \frac{1}{\delta^{*3}}\right) \tag{48}$$

$$\frac{d\Delta P^*}{d\xi^*} = -\frac{D_2 \dot{Q}^*}{\delta^{*3}} \tag{49}$$

$$\frac{d\dot{Q}^*}{d\xi^*} = \frac{1 - \Delta p^*}{\delta^* + D_3}, \tag{50}$$

where the dimensionless constants D have the following form:

$$D_1 = \frac{\delta_{ad} \Delta p_{ad}}{\sigma}, \quad D_2 = \frac{3\mu_1 k_l \left(T_{w,mic} - T_{sat}\right)}{\rho_l h_{fg} {\delta_{ad}}^2 \Delta p_{ad}}, \quad D_3 = \frac{k_l R_{int}}{\delta_{ad}}. \tag{51}$$

The Runge-Kutta method [71] was used to solve the system of ODEs based on the dimensionless boundary conditions given by:

$$\delta^*_{ad} = 1 + \varepsilon_1, \quad \delta'^*_{ad} = 0, \quad \Delta p^*_{ad} = 1, \quad Q^*_{ad} = 0 + \varepsilon_2. \tag{52}$$

The initial boundary values in the microlayer model correspond to the magnitude of the variables at the adsorbed layer. At the adsorbed layer, the heat flux is zero since the evaporative mass flux is null. In addition, the adsorbed layer consists of a flat interface with a zero interface-curvature. Under these conditions, the initial boundary values are given by:

$$\delta_{ad} = \left(\frac{A}{\rho_l h_{fg} \left(\frac{T_{w,mic}}{T_{sat}} - 1 \right)} \right)^{1/3} \tag{53}$$

$$\delta_{ad}{}' = 0 \tag{54}$$

$$\Delta P_{ad} = \frac{A}{{\delta_{ad}}^3} \tag{55}$$

$$\dot{Q}_{ad} = 0 \tag{56}$$

where Eq. (53) comes from Eq. (40) and the assumption of zero heat flux at the adsorbed layer, Eq. (54) implies a zero interface curvature at the adsorbed layer, Eq. (55) comes from Eq. (42) and the assumption of zero interface curvature, and Eq. (56) assumes a zero heat flux at the adsorbed layer (no evaporation occurs at the adsorbed layer).

The solution of the microloyer model requires a small perturbation of the dimensionless microlayer profile (ε_1) and of the integrated heat flux (ε_2) to avoid the trivial solution. The solution considered ε_1=0.002 and the value of ε_2 was adjusted depending on the microlayer slope. Initially, the length of integration is small (0.01 μm). The procedure to solve the microlayer model is as follows: (i) adjust ε_2 to have a maximum dimensionless slope of 1, (ii) increase the length of integration until the maximum dimensionless slope is 10, (iii) return to step (i) with the new integrated microlayer length. The simulation defined the total microlayer length ξ_{mic} as the location where the microlayer heat flux becomes minimal and

the interface slope becomes asymptotic.

The condition of equal apparent contact angle φ at the end of the transition region and at the beginning of the macroregion (given by sharp interface reconstruction with the liquid volume-fractions) couples the microlayer model to the general simulation. At the transition region, the effect of adhesion forces and interface curvature on the liquid evaporation are minimal. However, significant evaporation appears at the transition region due to the small distance between the interface (with a temperature T_{sat}) and the heated surface (with a temperature T_w) and to the length of the transition region (significantly larger relative to the microlayer length). As in Kunklemann and Stephan [10], the simulation in the present work assumes 1D heat transfer in the transition region. Under these conditions, the integration of Eq. (57) gives the integrated heat flux in the transition region:

$$\dot{Q}_{tran} = \int_{\xi_{mic}}^{\xi_{cl}} q_e{}'' d\xi = \frac{k_l \left(T_w - T_{sat}\right)}{\tan(\varphi)} \ln\left(1 + \frac{1\left(\xi_{cl} - \xi_{mic}\right)\tan(\varphi)}{\delta_{mic} + k_l R_{int}}\right). \tag{57}$$

The total heat at the contact line is the sum of the integrated heat fluxes at the microlayer and transition regions as given by:

$$Q_{cl} = \left(\dot{Q}_{mic} + \dot{Q}_{tran}\right) l_{cl}, \tag{58}$$

where l_{cl} is the perimeter of the contact-line. Eq. (59) gives the mass source term that accounts for the contact line evaporation in the VOF tracking interface equation. The simulation defines a mass source term at the computational cell of volume V_c that has the contact line. The magnitude of the mass source term is given by:

$$S_{mic} = \frac{Q_{cl}}{h_{fg}} \frac{1}{V_c}. \tag{59}$$

3.3 ANSYS-Fluent Customization

The numerical simulation of interfacial phase change was performed in ANSYS-Fluent, which solves the governing equations for mass, momentum (including surface tension effects), and energy conservation. In addition, the numerical package solves the Volume-of-Fluid (VOF) tracking interface equation. However, we developed User-Defined-Functions (UDFs) to modify the software; the modifications account for the interfacial transport including: (i) estimation of the mass transfer with interfacial temperature gradients (see section 2.1), (ii) interpolation and fixing of the interface-cells temperature to include a sharp interface with a saturation temperature (see section 2.2), (iii) declaration of mass transfer only at mass-transfer-cells to avoid the deformation of the interface (see section 2.3).

Figure 18 shows a flow diagram of the procedures in the simulation; the simulation starts with initial conditions for VOF and temperature. UDFs fix the temperature of the interface-cells before the software solves the governing equations for momentum and energy conservation. The software iterates until the convergence criteria is satisfied (the convergence criteria is $1x10^{-6}$ for all the equations). After the convergence is satisfied, UDFs estimate the mass source term with the temperature gradient at the interface. After UDFs declare the mass source, the software advances the interface by solving the VOF equation. In the next time step, the process repeats with the new interface position.

In the simulation, the UDF DEFINE-INIT sets the initial conditions of volume-fraction and temperature field. The UDF DEFINE-ADJUSTS was used to identify the interface-cells, the neighboring-interface-cells and the G-cells. The identification of the cells is needed to compute mass transfer, to interpolate the temperature of the interface-cells, and to declare mass transfer only at the mass-transfer-cells.

The computation of the mass source terms with the proposed external-software method (see section 2.1.3) requires the coupling of ANSYS-Fluent and MATLAB. The UDF DEFINE-EXECUTE-AT-THE-END writes and reads a text file at the end of each time step. The procedure to write and read data is as follows:

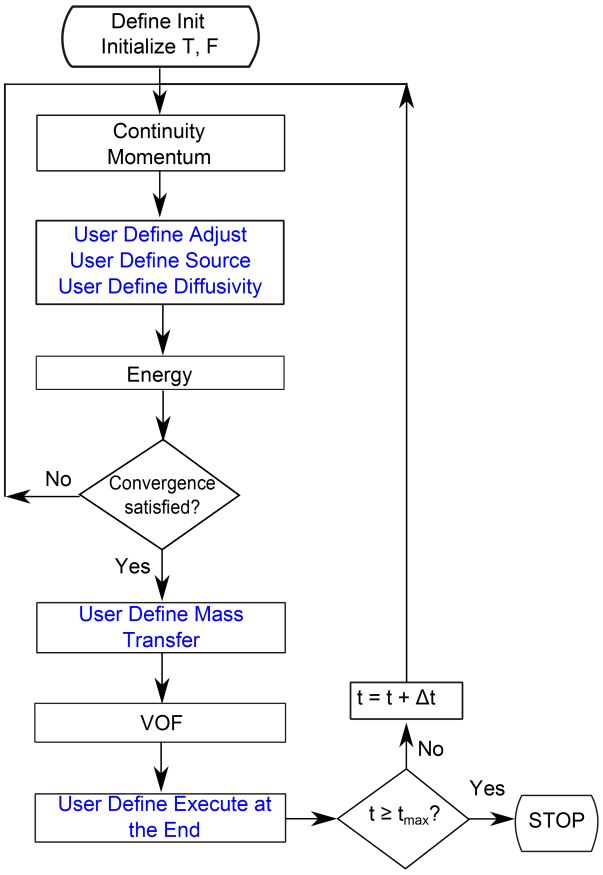

Figure 18: ANSYS-Fluent procedures and customization.

i *c-loop* visits the computational cells near the interface and macros get the information needed to perform the interpolation (coordinates of reference points, and coordinates and temperatures of the computational cells near the interface). The macro *fprintf* stores the values into the text file.

ii MATLAB continuously reads the files generated by Fluent and interpolates temperatures into a text file. An infinite while cycle in MATLAB reads and writes the text files continuously. The macro *textscan* reads the text file and the macro *fprintf* stores the interpolated temperature into a text file. MATLAB computes the mass transfer and writes the values into a text file. In MATLAB, Eq. (14) gives the interfacial temperature gradients.

iii *c-loop* visits the computational cells and macros retrieve the information of the interfa-

cial temperature gradients to the corresponding mass-transfer-cells. The UDM0 stores the values of the interfacial temperature gradients, and the UDM1 stores the values of the mass source terms.

The UDF DEFINE-MASS-TRANSFER reads UDM1 and declares mass source terms in the VOF equation.

The fixing and interpolation of the interface-cell temperature requires the declaration of source terms in a User-Defined-Scalar (UDS). The UDF DEFINE-ADJUST computes the source terms after each iteration. The procedure is as follows:

i *c-loop* reads UDM0 that has the interfacial temperature gradients at each mass-transfer-cell.

ii Eq. (18) computes the interface-cell temperature.

iii Eq. (20) computes the source terms. The UDF stores the source term in UDM2.

The UDF DEFINE-SOURCE reads UDM2 and declares source terms in the UDS equation.

The solution of a UDS provides the temperature distribution in the simulation. In the simulation of spherical bubble growth and nucleate boiling, the UDF DEFINE-DIFFUSIVITY assigns a diffusivity of zero on vapor cells, and a diffusivity of water on interface-cells and liquid cells. The UDF DEFINE-UDS-UNSTEADY assigns a liquid density of water on interface-cells and liquid cells. Liquid properties on interface-cells allow the discretization of the neighboring-mass transfer cells with liquid properties instead of average properties. In addition, liquid properties on interface-cells eliminate the strong numerical diffusion that comes with the use average properties on interface-cells.

The simulation declares mass transfer just on mass-transfer-cells. UDMs classify the computational cells. UDFs assign UDM3=1 to mass-transfer-cells and UDM3=2 to neighboring-mass-transfer-cells. The identification of the cells is as follows:

i *f-loop* visits the faces of cells in the computational domain.

ii Connectivity macros retrieve the volume-fractions of two consecutive cells c0 and c1.

iii If VOFc0-VOFc1 is higher than zero and VOFc1 is equal to zero, then UDM3c0 is equal to one (the cell c0 is a mass-transfer-cell). Similarly, if VOFc0-VOFc1 is higher than zero, UDM3c1 equals one, and UDM3c0 is different to one, then UDM3c0 is equal to two (the cell c0 is a neighboring-mass-transfer-cell).

The UDF EXECUTE-AT-THE-END cleaned the variables, including the UDMs that passed information between UDFs.

To the knowledge of the present authors, this is the first work that uses ANSYS-Fluent to simulate nucleate boiling where the mass transfer depends on the interfacial temperature gradients. This is possible since the developed models are compatible with traditional discretization schemes used in numerical packages.

4 Theoretical Cases

A fundamental understanding of the phase change process in a nucleating bubble can be obtained by first considering evaporation at a planar liquid-vapor interface or the growth of spherical bubbles in superheated liquid. Planar interface models focus on the driving mechanisms of phase change such as temperature distributions, latent heat absorption, interface motion, and vapor expansion due to density difference between liquid and vapor phases. Spherical bubble growth considers effect of interface curvature. In numerical modeling, planar interface models provide a simplified scenario to test the effectiveness of the numerical method to capture the interfacial processes, whereas spherical bubble growth test the accuracy of the model on computing mass transfer and surface tension effects with variable interface inclinations in a three dimensional environment. The following subsections describe the theoretical framework of these problems.

4.1 Planar Interface Evaporation

Stefan formulated the relation between temperature distribution and mass transfer across the interface between two phases to predict ice growth in polar seas [72]. Figure 19 shows the model considered by Stefan. As a first approximation, Stefan assumed a linear temperature profile within the ice, which is a reasonable assumption since that the ice is unable to store energy due to its low specific heat. However, the same work derived exact solutions where the heat conduction equation dictated the ice temperature. Data from a German expedition to the polar seas taken during first winter months of 1831 showed a linear temperature profile, the author also noted that during later winter months (March to May) the temperature gradients change with the depth (non linear temperature profile). Stefan pointed out that during ice formation the surface temperature changes with time; the surface temperature is initially zero, gradually decrease to a minimum value (during first winter months), and then increases to zero again (during later winter months). Therefore, the author formulated two

cases: (i) constant surface temperature, and (ii) variation of surface temperature with time. Results of case (ii) where compared with measurements made in the winter 1831-1832 at the Gulf of Baoothia. The comparison indicated that the prediction of the ice thickness was acceptable for first winter months with relative errors of 0.92%.

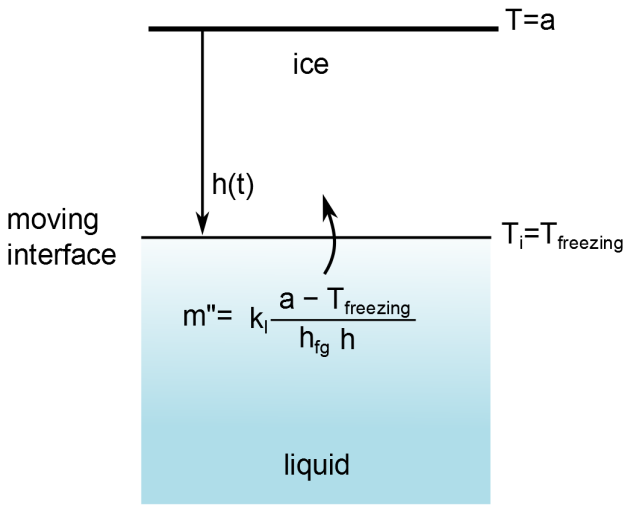

Figure 19: Schematic diagram of model to predict ice formation considered by Stefan.

Numerical simulations of boiling have used Stefan type problems to validate heat and mass transfer models at the interface level (the validation compares the numerical results against the theoretical data). However, it is common to adopt Stefan problems that consider heat transfer only on one phase that either neglect or include density effects. Juric and Tryggvason [8] simulated superheated liquid in contact with the surface where the interface moved towards the surface; the assumption of a linear temperature profile on the liquid gave the interface displacement. Son and Dhir [16] simulated superheated vapor in contact with the surface (Stefan problem); the solution of the 1D heat conduction equation gave the interface displacement. Welch and Wilson [40] simulated Stefan problem. In addition, the authors proposed the case of saturated vapor in contact with the surface and superheated liquid (sucking problem); a semianalytical solution provided the interface displacement and the change on the liquid temperature with time. Hardt and Wondra [9] simulated Stefan problem; in addition, the authors simulated superheated liquid in contact with the heated

surface where the interface moved towards the surface (no comparison with theoretical solutions was provided). Kunkelmann and Stephan [10] and Sato and Niceno [3] simulated Stefan problem in boiling and sucking problem to validate a simulation of nucleate boiling; the semianalytical solution provided by Welch and Wilson [40] gave the theoretical data of sucking problem.

Previous models have in general relied on the basic case of evaporation from saturated vapor to superheated liquid separated by a planar or curved interface. However, different liquid and vapor conditions are employed in simulating complex evaporation/boiling processes. This section conveys eight different cases classified under the Stefan problem that employ different combinations of a saturated, subcooled, or superheated liquid phase and a saturated or superheated vapor phase along with the same or different phase densities. This section presents theoretical equations for these proposed cases. These cases are recommended for validating numerical codes developed for simulating the evaporation/boiling processes. The theoretical work of Alexiades and Solomon for melting and freezing processes [73] served as a platform to derive the theoretical solutions.

Figure 20 shows the various configurations of interfacial evaporation on a planar interface proposed by the present work. Details on the governing physics for these cases will be provided in the following sections. Briefly, the configurations are divided in one-phase Stefan problems and two-phase Stefan problems. In the one-phase Stefan problems, there is heat transfer only on one phase whereas the other phase remains at the saturation temperature. Case Sup-V/Sat-L has superheated vapor and saturated liquid, and case Sat-V/Sup-L has saturated vapor and superheated liquid. In the two-phase Stefan problems there is heat transfer on both phases. Case Sup-V/Sub-L has superheated vapor and subcooled liquid, and case Sup-V/Sup-L has both vapor and liquid phases in superheated state. Each of these problems either neglects or considers the density effects (the two phases can have similar or different density values). Table 1 shows the abbreviations used to identify the different configurations. Sup and Sub stand for superheated and subcooled, respectively. V and L

stand for vapor and liquid, respectively. The cases that are abbreviated with an addition of ρ have a liquid-to-vapor density ratio that is higher than one.

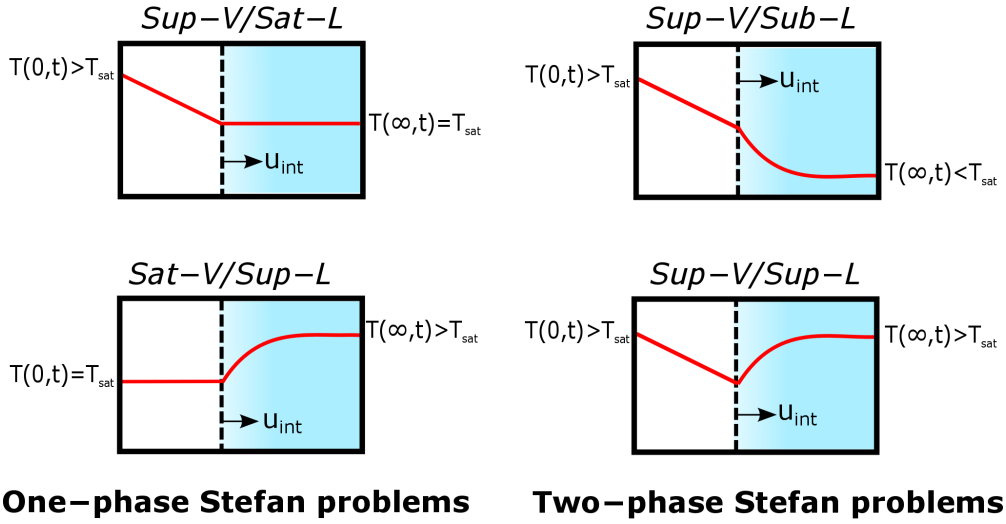

Figure 20: Various cases of planar interface evaporation. The phase with blue color is liquid and the phase with white color is vapor. The red line indicates the temperature profile in each phase. Each case either includes or neglects the density effects.

4.1.1 One Phase Stefan problems with Sup-V/Sat-L and Sup-V/Sat-L-ρ

Figure 21 shows a schematic diagram of the Stefan problem with Sup-V/Sat-L and Sup-V/Sat-L-ρ. Initially, the interface is at $x = 0$ m, and the domain has only liquid with the saturation temperature $T(x,t) = T_{sat}$ (saturated liquid). The surface has a temperature $T(0,t) = T_A > T_{sat}$, which introduces heat into the vapor phase. The heat that arrives to the interface evaporates the liquid in contact with the interface, which makes the interface to move in the x-direction. The amount of liquid that evaporates (evaporative mass flux) depends on the heat flux at the interface on vapor side and the latent heat of evaporation. The liquid properties have no effect on the interface displacement and temperature distribution since the liquid remains at a constant temperature (saturation temperature). Therefore, Sup-V/Sat-L and Sup-V/Sat-L-ρ have the same theoretical formulation and solution.

The theoretical formulation of Sup-V/Sat-L and Sup-V/Sat-L-ρ predicts the interface displacement and the liquid temperature distribution along time. The equation that governs

Table 1: Classification of the planar configurations proposed for validation of boiling simulations.

Proposed Eight Stephan Problems	**Abbr.**
One-phase without density difference	
Superheated-Vapor/Saturated-Liquid	Sup-V/Sat-L
Saturated-Vapor/Superheated-Liquid	Sat-V/Sup-L
Two-phase without density difference	
Superheated-Vapor/Subcooled-Liquid	Sat-V/Sub-L
Superheated-Vapor/Superheated-Liquid	Sup-V/Sup-L
One-phase with density difference	
Superheated-Vapor-ρ_v/Saturated-Liquid-ρ_l	Sup-V/Sat-L-ρ
Saturated-Vapor-ρ_v/Superheated-Liquid-ρ_l	Sat-V/Sup-L-ρ
Two-phase with density difference	
Superheated-Vapor-ρ_v/Subcooled-Liquid-ρ_l	Sat-V/Sub-L-ρ
Superheated-Vapor-ρ_v/Superheated Liquid-ρ_l	Sup-V/Sup-L-ρ

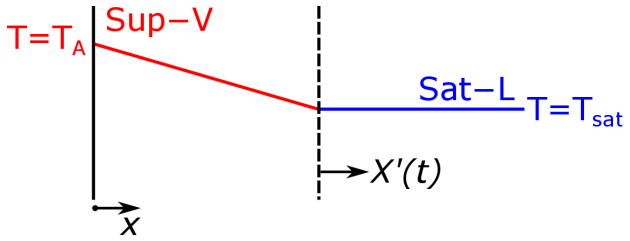

Figure 21: Stefan problem with superheated vapor and saturated liquid (Sup-V/Sat-L).

the heat transfer on the vapor is the 1D heat conduction equation:

$$\frac{\partial T}{\partial t} = \alpha_v \frac{\partial^2 T}{\partial x^2}, \quad x \leq X(t), \tag{60}$$

where α_v is the vapor thermal diffusivity. Eq. (60) is valid in the interval $x \leq X(t)$, where $X(t)$ is the interface position at time t.

The boundary conditions consider a temperature $T_A > T_{sat}$ at the surface and a temperature T_{sat} at the interface:

$$T\left(0,t\right) = T_A \tag{61}$$

$$T\left(X(t),t\right) = T_{sat}. \tag{62}$$

Initially, the interface is at $x = 0$ and the domain has only saturated liquid with a temperature T_{sat}. Eqs. (63)-(64) provide the initial conditions:

$$X(0) = 0 \tag{63}$$

$$T(x,0) = T_{sat}. \tag{64}$$

Eq. (65) gives the relation between the interface velocity $X'(t)$ and the interfacial temperature gradients:

$$m_e{}'' = X'(t)\rho_v = \frac{q_{int}{}''}{h_{fg}} = \frac{k_v}{h_{fg}} \frac{dT}{dx}_{int,v}. \tag{65}$$

The above formulation describes a boundary moving problem. The exact solution can be found through a similarity variable [73]. Eq. (66) gives the interface displacement, Eq. (67) gives the temperature distribution on the vapor (where erf is the error function), and Eq. (68) gives the constant β_v:

$$X(t) = \beta_v \sqrt{t} \tag{66}$$

$$T(x,t) = T_A - (T_A - T_{sat}) \frac{\text{erf}\left(\frac{x}{2\sqrt{\alpha_v t}}\right)}{\text{erf}\left(\frac{\beta_v}{2\sqrt{\alpha_v}}\right)}, \quad x \leq X(t) \tag{67}$$

$$\frac{\beta_v}{2\sqrt{\alpha_v}} = \frac{c_v (T_A - T_{sat})}{h_{fg}\sqrt{\pi}\exp\left(\frac{{\beta_v}^2}{4\alpha_v}\right)\text{erf}\left(\frac{\beta_v}{2\sqrt{\alpha_v}}\right)}. \tag{68}$$

Eq. (68) is used to find the constant β_v; the procedure to find β_v consists of an iterative numerical method.

4.1.2 One Phase Stefan problem with Sat-V/Sup-L

Figure 22 shows a schematic diagram of the Stefan problem with Sat-V/Sup-L. Initially, the interface is at $x = 0$ m, and the domain has only liquid with a temperature $T(x,t) = T_B > T_{sat}$ (superheated liquid). The problem assumes that the interface has the saturation temperature T_{sat}. Due to the temperature difference, the interface cools down the superheated

liquid, which creates a thermal film on the liquid in contact with the interface. Heat conduction increases the thickness of the thermal film with time. Moreover, the liquid in contact with the interface converts into vapor (evaporation), which makes the interface to move in the x-direction. The amount of liquid that evaporates (interfacial mass flux) depends on the heat flux at the interface on the liquid side and the latent heat of evaporation.

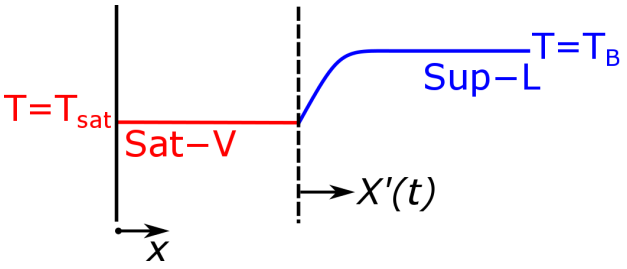

Figure 22: Stefan problem with saturated vapor and superheated liquid (Sat-V/Sup-L).

The theoretical formulation of Sat-V/Sup-L predicts the interface displacement and the liquid temperature distribution along time. The equation that governs the heat transfer on the liquid is the 1D heat conduction equation:

$$\frac{\partial T}{\partial t} = \alpha_l \frac{\partial^2 T}{\partial x^2}, \quad X(t) \leq x \leq \infty, \tag{69}$$

where α_l is the liquid thermal diffusivity. Eq. (69) is valid in the interval $X(t) \leq x \leq \infty$, where $X(t)$ is the interface position at time t, and ∞ is a long distance from the interface (a distance such that the boundary has no effect on the temperature of the liquid). Eqs. (70)-(71) give the boundary conditions:

$$T\left(X(t), t\right) = T_{sat} \tag{70}$$

$$T\left(\infty, t\right) = T_B. \tag{71}$$

Initially, the interface is at $x = 0$ and the domain has only superheated liquid with a

temperature T_B. Eqs. (72)-(73) provide the initial conditions:

$$X(0) = 0 \tag{72}$$

$$T(x,0) = T_B. \tag{73}$$

Eq. (74) gives the relation between the interface velocity $X'(t)$ and the interfacial temperature gradients:

$$m_e{}'' = X'(t)\rho_v = \frac{q_{int}{}''}{h_{fg}} = \frac{k_l}{h_{fg}} \frac{dT}{dx}_{int,l}. \tag{74}$$

The above formulation describes a boundary moving problem. The exact solution can be found through a similarity variable [73]. Eq. (75) gives the interface displacement, Eq. (76) gives the temperature distribution on the liquid (where erfc is the complementary error function), and Eq. (77) gives the constant β_l.

$$X(t) = \beta_l \sqrt{t} \tag{75}$$

$$T(x,t) = T_B - (T_B - T_{sat}) \frac{\text{erfc}\left(\frac{x}{2\sqrt{\alpha_l t}}\right)}{\text{erfc}\left(\frac{\beta_l}{2\sqrt{\alpha_l}}\right)}, \quad x \geq X(t) \tag{76}$$

$$\frac{\beta_l}{2\sqrt{\alpha_l}} = \frac{Ja}{\sqrt{\pi}\exp\left(\frac{{\beta_l}^2}{4\alpha_l}\right)\text{erfc}\left(\frac{\beta_l}{2\sqrt{\alpha_l}}\right)}, \tag{77}$$

where Ja is the Jakob number given by:

$$Ja = \frac{\rho_l c_l (T_B - T_{sat})}{\rho_v h_{fg}}, \tag{78}$$

the Jakob number represents the ratio of sensible heat in the system to the latent heat due to interfacial evaporation.

4.1.3 One Phase Stefan problem with Sat-V/Sup-L-ρ

Stefan problem with saturated vapor and superheated liquid with density difference Sat-V/Sup-L-ρ is similar to Sat-V/Sup-L. However, in Sat-V/Sup-L both phases (vapor and liquid) have the same density, whereas in Sat-V/Sup-L-ρ the phases have a different density. Due to mass conservation $\rho_l \Delta V_l = \rho_v \Delta V_v$ where, ΔV_l and ΔV_l are the volume of liquid consumed and the volume of vapor generated in a time interval, respectively. In the case of water, the density ratio between the liquid vapor is relatively large $\rho_l/\rho_v \approx 1,600$. Therefore, for a small volume of liquid consumed a large volume of vapor is generated during the evaporation of water. Such vapor expansion sends the liquid outwards along the x-direction. The liquid motion creates a convective heat transfer on the liquid that moves the liquid thermal film with the interface.

In Sat-V/Sup-L-ρ, 1D heat conduction-advection equation predicts the heat transfer on the liquid as given by:

$$\frac{\partial T}{\partial t} + u_l \frac{\partial T}{\partial x} = \alpha_l \frac{\partial^2 T}{\partial x^2}, \quad X(t) \leq x \leq \infty, \tag{79}$$

where u_l is the liquid velocity due to the vapor expansion. Eqs. (70)-(71) give the boundary conditions, Eqs. (72)-(73) give the initial condition, and Eq. (74) gives the relation between the interface velocity $X'(t)$ and the interfacial temperature gradients.

The solution of the above formulation requires to find a relation between the liquid velocity u_l and the interface velocity X'. The condition of mass conservation provides such a relation. Eq. (80) gives the mass of vapor generated in an interval of time Δt, where A_i is the interface surface area. Similarly, Eq. (81) gives the mass of liquid generated in an interval of time Δt:

$$m_v = A_i X'(t) \rho_v \Delta t \tag{80}$$

$$m_l = A_i \left(X'(t) - u_l \right) \rho_l \Delta t, \tag{81}$$

the condition of mass conservation $m_v = m_l$ gives the relation between the liquid velocity and the interface velocity as:

$$u_l = \left(1 - \frac{\rho_v}{\rho_l}\right) X'(t). \tag{82}$$

Eq. (82) indicates that the liquid velocity is zero if both liquid and vapor phases have the same density. If the two densities are equal, the liquid velocity is zero since the volume of vapor generated is the same as the volume of liquid consumed. However, small density ratios ρ_v/ρ_l generate large liquid velocities due to the vapor expansion.

The exact solution of Eq. (79) combined with Eq. (79) and Eq. (82) can be found through a similarity variable [73]. Eq. (83) gives the interface displacement, Eq. (84) gives the temperature distribution on the liquid, and Eq. (85) gives the constant β_{l2}:

$$X(t) = \beta_{l2}\sqrt{t} \tag{83}$$

$$T(x,t) = T_B - (T_B - T_{sat}) \frac{\operatorname{erfc}\left(\frac{x}{2\sqrt{\alpha_l t}} - \frac{\varepsilon\beta_{l2}}{2\sqrt{\alpha_l}}\right)}{\operatorname{erfc}\left(\frac{\beta_{l2}}{2\sqrt{\alpha_l}}\frac{\rho_v}{\rho_l}\right)}, \quad x \geq X(t) \tag{84}$$

$$\frac{\beta_{l2}}{2\sqrt{\alpha_l}} = \frac{\mathrm{Ja}}{\sqrt{\pi}\exp\left(\frac{{\beta_{l2}}^2}{4\alpha_l}\left(\frac{\rho_v}{\rho_l}\right)^2\right)\operatorname{erfc}\left(\frac{\beta_{l2}}{2\sqrt{\alpha_l}} - \frac{\varepsilon\beta_{l2}}{2\sqrt{\alpha_l}}\right)}, \tag{85}$$

where $\varepsilon = 1 - \rho_v/\rho_l$. It is important to observe that the Eqs. (83)-(85) for Sat-V/Sup-L-ρ with $\rho_l \gg \rho_v$ reduce to Eqs. (75)-(77) for Sat-V/Sup-L with $\rho_l = \rho_v$. In addition, it should be noted that Eq. (85) is used to find the constant β_{l2}; the procedure to find β_v consists of an iterative numerical method.

4.1.4 Two-phase Stefan problems with no density effect

Figure 23 shows a schematic diagram of two-phase Stefan problems. Figure 23(a) shows a two-phase Stefan problem with superheated vapor and subcooled liquid (Sup-V/Sub-L), and Figure 23(b) shows a two-phase Stefan problem with superheated vapor and superheated liquid (Sup-V/Sup-L). Sup-V/Sub-L considers that the liquid temperature is smaller than

the saturation temperature, whereas Sup-V/Sup-L considers that the liquid temperature is higher than the saturation temperature. Both problems assume that the vapor is superheated and that the interface has the saturation temperature T_{sat}. The interaction between the interface and the liquid forms a thermal film on the liquid. Heat conduction increases the thickness of the liquid thermal film with time.

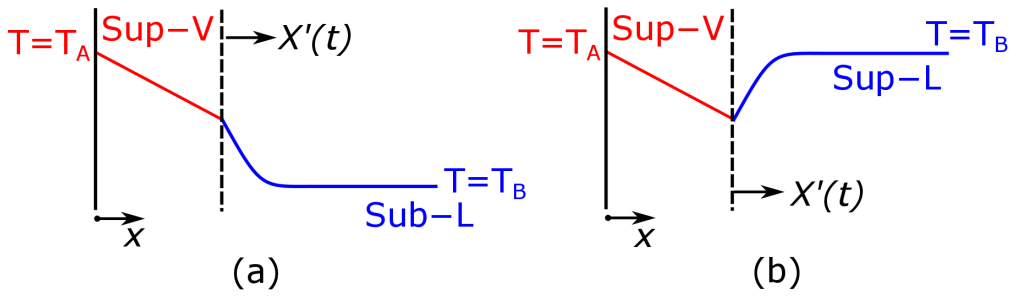

Figure 23: Two-phase Stefan problems: (a) Superheated vapor and subcooled liquid (Sup-V/Sub-L), (b) superheated vapor and superheated liquid (Sup-V/Sup-L).

The amount of liquid that evaporates depends on the net heat flux at the interface. In Sup-V/Sub-L, the heat travels from the vapor to the interface, and from the interface to the liquid; therefore, the heat from the vapor phase evaporates liquid, whereas the heat from the liquid phase condenses vapor. In Sup-V/Sub-L, both phases transfer heat to the interface; therefore, the heat from both phases evaporates liquid.

The theoretical formulation of two-phase Stefan problems predicts the interface displacement and the vapor and liquid temperature distribution along time. Heat conduction Eqs. (86) and (87) govern the heat transfer on the vapor and on the liquid:

$$\frac{\partial T}{\partial t} = \alpha_v \frac{\partial^2 T}{\partial x^2}, \quad 0 \leq x \leq X(t) \tag{86}$$

$$\frac{\partial T}{\partial t} = \alpha_l \frac{\partial^2 T}{\partial x^2}, \quad X(t) \leq x \leq \infty. \tag{87}$$

It is important to highlight that convection heat transfer on the liquid is null since both

phases have the same density. Eqs. (88)-(90) give the boundary conditions:

$$T(0,t) = T_A \tag{88}$$

$$T(X(t),t) = T_{sat} \tag{89}$$

$$T(\infty,t) = T_B. \tag{90}$$

Initially, the interface is at $x = 0$ and the domain has only with a temperature T_B. Sup-V/Sub-L has $T_B < T_{sat}$, whereas Sup-V/Sup-L has $T_B > T_{sat}$. Eqs. (72)-(73) describe the initial conditions:

$$X(0) = 0 \tag{91}$$

$$T(x,0) = T_B. \tag{92}$$

Eq. (93) provides the relation between the interface velocity $X'(t)$ and the interfacial temperature gradients. The evaporative mass flux depends on the sum of the heat fluxes at the interface given by:

$${m_e}'' = X'(t)\rho_v = \frac{{q_{int}}''}{h_{fg}} = \frac{-k_v}{h_{fg}}\frac{dT}{dx}_{int,v} + \frac{k_l}{h_{fg}}\frac{dT}{dx}_{int,l}, \tag{93}$$

where $\frac{dT}{dx}_{int,v}$, and $\frac{dT}{dx}_{int,l}$ are the temperature gradients at the interface on the vapor and liquid side, respectively. The above formulation describes a boundary moving problem. The exact solution can be found through a similarity variable [73]. Eq. (94) gives the interface displacement, Eq. (95) gives the temperature distribution on the vapor, Eq. (96) gives the

temperature distribution on the liquid, and Eq. (97) gives the constant β_{vl}:

$$X(t) = \beta_{vl}\sqrt{t} \tag{94}$$

$$T(x,t) = T_A - (T_A - T_{sat})\frac{\operatorname{erf}\left(\frac{x}{2\sqrt{\alpha_v t}}\right)}{\operatorname{erf}\left(\frac{\beta_{vl}}{2\sqrt{\alpha_v}}\right)}, \quad 0 \le x \le X(t) \tag{95}$$

$$T(x,t) = T_B - (T_B - T_{sat})\frac{\operatorname{erfc}\left(\frac{x}{2\sqrt{\alpha_l t}}\right)}{\operatorname{erfc}\left(\frac{\beta_{vl}}{2\sqrt{\alpha_l}}\right)}, \quad X(t) \le x \le \infty \tag{96}$$

$$\frac{\beta_{vl}\sqrt{\pi}h_{fg}}{2} = \frac{\sqrt{\alpha_v}c_v\left(T_A - T_{sat}\right)}{\exp\left(\frac{{\beta_{vl}}^2}{4\alpha_v}\right)\operatorname{erf}\left(\frac{\beta_{vl}}{2\sqrt{\alpha_v}}\right)} + \frac{\sqrt{\alpha_l}c_l\left(T_B - T_{sat}\right)}{\exp\left(\frac{{\beta_{vl}}^2}{4\alpha_l}\right)\operatorname{erfc}\left(\frac{\beta_{vl}}{2\sqrt{\alpha_l}}\right)}, \tag{97}$$

Eq. (94) indicates that the constant β_{vl} gives the interface displacement, and Eq. (97) shows that the β_{vl} depends on the temperature difference on the vapor and liquid sides. The only unknown parameter in Eq. (97) is the constant β_{vl}; a numerical method was used to find the constant β_{vl}.

4.1.5 Two-phase Stefan problems with density effect

As described in section 4.1.4, phase change between two phases of different density gives rise to an expansion of the phase with the lower density. In the proposed Stefan problems, the vapor expansion sends the liquid outwards along the x-direction. The liquid expansion creates a convective effect on the liquid that moves the heat with the interface.

In two-phase Stefan problems with density effect, Eq. (98) governs the heat transfer on the vapor and Eq. (99) governs the heat transfer on the liquid. The extra term in Eq. (99) comes from the convective effect that arises with the vapor expansion.

$$\frac{\partial T}{\partial t} = \alpha_v \frac{\partial^2 T}{\partial x^2}, \quad 0 \le x \le X(t) \tag{98}$$

$$\frac{\partial T}{\partial t} + u_l \frac{\partial T}{\partial x} = \alpha_l \frac{\partial^2 T}{\partial x^2}, \quad X(t) \le x \le \infty. \tag{99}$$

Eqs. (88)-(90) give the boundary conditions, Eqs. (91)-(92) give the initial condition, and

Eq. (93) gives the relation between the liquid velocity $X'(t)$ and the interfacial temperature gradients.

Mass conservation, as given by Eq. (82), provides a relation between the liquid velocity u_l and the interface velocity X'. The exact solution for the interface displacement and the temperature distribution on both phases can be found through a similarity variable [73]. Eq. (100) gives the interface displacement, Eq. (101) gives the temperature distribution on the vapor, Eq. (102) gives the temperature distribution on the liquid, and Eq. (103) gives the constant β_{vl2}:

$$X(t) = \beta_{vl2}\sqrt{t} \tag{100}$$

$$T(x,t) = T_A - (T_A - T_{sat})\frac{\operatorname{erf}\left(\frac{x}{2\sqrt{\alpha_v t}}\right)}{\operatorname{erf}\left(\frac{\beta_{vl2}}{2\sqrt{\alpha_v}}\right)}, \quad 0 \leq x \leq X(t) \tag{101}$$

$$T(x,t) = T_B - (T_B - T_{sat})\frac{\operatorname{erfc}\left(\frac{x}{2\sqrt{\alpha_l t}} - \frac{\varepsilon\beta_{vl2}}{2\sqrt{\alpha_l}}\right)}{\operatorname{erfc}\left(\frac{\beta_{vl2}}{2\sqrt{\alpha_l}}\frac{\rho_v}{\rho_l}\right)}, \quad X(t) \leq x \leq \infty \tag{102}$$

$$\frac{\beta_{vl2}\sqrt{\pi}h_{fg}}{2} = \frac{\sqrt{\alpha_v}c_v\left(T_A - T_{sat}\right)}{\exp\left(\frac{{\beta_{vl2}}^2}{4\alpha_v}\right)\operatorname{erf}\left(\frac{\beta_{vl2}}{2\sqrt{\alpha_v}}\right)} + \frac{\sqrt{\alpha_l}c_l\left(T_B - T_{sat}\right)}{\exp\left(\frac{{\beta_{vl2}}^2}{4\alpha_l}\left(\frac{\rho_v}{\rho_l}\right)^2\right)\operatorname{erfc}\left(\frac{\beta_{vl2}}{2\sqrt{\alpha_l}}\frac{\rho_v}{\rho_l}\right)}, \tag{103}$$

where $\varepsilon = 1 - \rho_v/\rho_l$. It is important to observe that the Eqs. (100)-(103) for two-phase Stefan problem with $\rho_l \gg \rho_v$ reduce to Eqs. (94)-(97) for two-phase Stefan problems with $\rho_l = \rho_v$.

4.2 Spherical Bubble Growth

The bubble growth process in uniformly superheated liquid has been well established in the technical literature. Plesset and Zwick [74] analyzed spherical bubble growth in a uniformly superheated liquid. The authors distinguished that initially bubble growth occurs due to inertia effects (inertia controlled region); in the inertia controlled region, the pressure difference at the interface is significant and the saturation temperature decreases. After a certain

time, the cooling effect at the interface governs the bubble growth rate (diffusion controlled region); in the diffusion controlled region, the pressure difference between the liquid and vapor is negligible and saturation temperature slightly changes. The evaporation changes the pressure inside the bubble, but the change in pressure is negligible. Therefore, the vapor expansion or the reduction in the saturation temperature due to change in pressure (refer to ideal gas law) is minimal and the bubble growth is mainly due to liquid evaporation. In the formulation of Plesset and Zwick [74], the momentum equation related the pressure difference at the interface with the interface velocity. An approximated equation found the interface temperature as a function of the bubble radius and interfacial heat flux. Temperature and pressure in the systems were related by a linear function. A source term added heat into the liquid to initiate bubble growth. The authors derived an expression to estimate the bubble radius and liquid temperature in the diffusion controlled region by assuming a limit in the interface temperature.

Mikic et al. [75] theoretically analyzed spherical bubble growth in the inertia-controlled and the diffusion-controlled regions. Mechanical energy conservation related vapor and liquid pressures with the interface velocity. The authors used an approximated equation to relate the saturation temperature to the bubble radius. Clausius-Clapeyron equation related the pressure difference with the temperature difference at the interface. The final solution predicted a different dependence of the bubble radius with time in each region: a linear dependence in the inertia-controlled region, and a square-root dependence in the diffusion-controlled region.

Scriven [76] analyzed spherical bubble growth in pure liquids and in binary solutions. Different to the theoretical models of Plesset and Zwick [74] and Mikic et al. [75], Scriven's model [76] finds the liquid temperature with the exact solution of the energy equation in spherical coordinates. The governing equations were energy and species for temperature and concentration distribution, respectively. Mass transfer model included normal gradients of concentration and temperature at the interface. The boundary conditions assumed a

constant interface concentration and temperature. The initial conditions had a small bubble immersed in an infinite mass of liquid with uniform temperature and composition. Initially, the liquid temperature was higher than the saturation temperature. The author pointed out that inertia and viscous effects are significant only during the early expansion of the bubble nucleus, and assumed similar pressures on vapor and liquid phases. The author derived exact solutions for bubble radius, temperature, and fluid composition in the diffusion-controlled region.

Numerical simulations of boiling use theoretical problems of Spherical bubble growth to validate the computation of surface tension, normal vectors, and mass transfer. Two different cases perform the validation: (i) adiabatic bubble growth where the evaporative mass flux is prescribed independently of the liquid temperature, and (ii) non-adiabatic bubble growth where the evaporative mass flux depends on the temperature gradients at the interface along the interface normal direction.

Adiabatic bubble growth evaluates errors on mass conservation and momentum transport. Tanguy et al. [32] simulated the evaporation of a two-dimensional droplet with a prescribed mass flux. Jump conditions included the interface effects in the Navier-Stokes equations. A projection method extended the liquid velocities into the gaseous phase and vice versa. The velocity fields were extended with a ghost pressure. Results showed that the computed temporal mass variation of the vaporizing drop and the exact solution were the same in a one milisecond interval. In a different work, Tanguy et al. [77] simulated adiabatic spherical bubble growth with a constant mass flux to compare whole domain and jump continuous formulations. The whole domain formulation smeared the interface across two or three computational cells, and the jump continuous formulation adopted ghost cells for the velocity. The simulation had a 2D axisymmetric domain with an initial bubble radius of 1 mm. Relative to the exact solution of the bubble radius, the whole domain formulation had 25% of error, and a continuous jump had less than 1% of error. The authors concluded that average velocities at the interface in the whole domain formulation and spurious currents

lead to wrong interface displacements.

Non-adiabatic bubble quantifies errors related to computation of mass transfer and solution of energy equation with a moving curved interface. Son [19] simulated spherical bubble growth with the level-set method. A volume correction step ensured mass conservation. The computed bubble maintained a spherical shape and the growth rate was equivalent to Scriven's type solution [76]. Tanguy et al. [77] simulated spherical bubble growth, the simulation accurately predicted growth rates relative to Scriven's theory with both a simple and a divergence-free velocity extrapolation. Relative to the linear extrapolation, a quadratic extrapolation of the thermal field improved the accuracy. Sato and Niceno [3] simulated a spherical bubble and compared their results against Scriven's theory [76]; the accuracy of the simulation improved with smaller cells (grid cell sizes of 1 μm or less). There are other works that have simulated non-adiabatic spherical bubble growth [10, 11].

Figure 24(a) shows a schematic diagram of adiabatic bubble growth. In adiabatic bubble growth, the evaporative mass flux is constant. The problem considers an initial small vapor bubble or radius R_0. The liquid in contact with the interface converts into vapor at a constant rate. The vapor expands due to the density difference between the liquid and vapor (as described in section 4.1.3), which sends the liquid outwards along the radial direction.

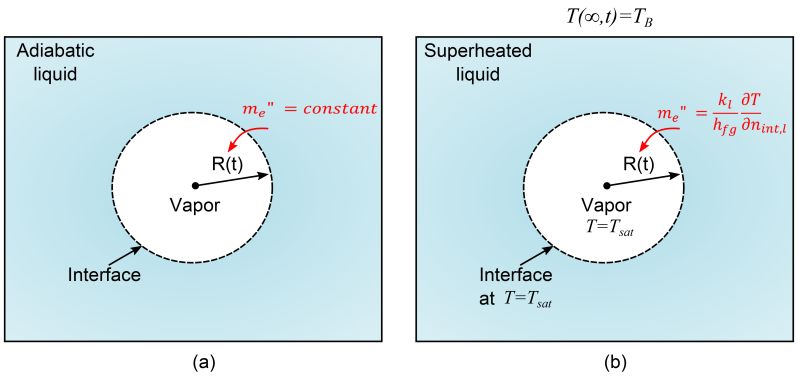

Figure 24: Spherical bubble growth problems: (a) adiabatic bubble growth with a constant mass flux, (b) non-adiabatic bubble growth with a mass flux that depends on interfacial temperature gradients.

A mass balance at the interface gives an expression for the growth of a spherical bubble

with a constant mass flux. The mass of liquid evaporated in a time interval is given by the evaporative mass flux times the surface area $m = m_e{}'' A_s dt$. At the same time, the mass of vapor generated is given by the interface displacement times the surface area and the vapor density $m = \rho_v A_s dr$. Therefore, with a constant mass flux (adiabatic bubble growth), the theoretical expression for the radius as a function of time is given by:

$$R(t) = R_0 + \frac{m_e{}''}{\rho_v} t, \tag{104}$$

where R_0 is the initial bubble radius, and $m_e{}''$ is the interfacial mass flux.

Figure 24(b) shows a schematic diagram of non-adiabatic bubble growth. In non-adiabatic bubble growth, the evaporative mass flux depends on the temperature gradients at the interface along the normal direction, which change with time due to the heat diffusion on the liquid. Initially, the interface is at $r = 0$ m, and the domain has only liquid with a temperature $T(r,t) = T_B > T_{sat}$ (superheated liquid). At zero seconds, the interface appears with a temperature T_{sat}. Due to the temperature difference, the interface cools down the superheated liquid, which creates a thermal film on the liquid in contact with the interface. Moreover, the liquid in contact with the interface converts into vapor (evaporation), which makes the interface to move in the radial direction. The vapor expands due to the density difference between the liquid and vapor (as described in section 4.1.4), which sends the liquid outwards along the radial direction.

In non-adiabatic bubble growth, 1D heat conduction-advection equation in spherical coordinates predicts the heat transfer on the liquid as shown by:

$$\frac{\partial T}{\partial t} + u_{l,r} \frac{\partial T}{\partial r} = \alpha_l \left(\frac{\partial^2 T}{\partial r^2} + \frac{2}{r} \frac{\partial T}{\partial r} \right), \quad R(t) \leq r \leq \infty, \tag{105}$$

where $u_{l,r}$ is the liquid velocity, which is function of the radial location. In Eq. (105), $R(t)$

is the bubble radius at time t. Eqs. (106)-(107) give the boundary conditions:

$$T\left(R(t),t\right) = T_{sat} \tag{106}$$

$$T\left(\infty,t\right) = T_B. \tag{107}$$

Initially, the bubble radius is zero and the domain has only superheated liquid with a temperature T_B. Eqs. (108)-(109) state the initial conditions:

$$R(0) = 0 \tag{108}$$

$$T\left(r,0\right) = T_B. \tag{109}$$

Eq. (110) gives the relation between the liquid velocity $R'\left(t\right)$ and the interfacial temperature gradients:

$$m_e{}'' = R'(t)\rho_v = \frac{q_{int}{}''}{h_{fg}} = \frac{k_l}{h_{fg}}\frac{dT}{dr}_{int,l} \tag{110}$$

Mass conservation, given in Eq. (82), provides the relation between the liquid velocity u_l at the interface and the interface velocity R'. The energy equation depends on the radial liquid velocity $u_{l,r}$. Therefore, the theoretical solution requires a relation between the radial liquid velocity $u_{l,r}$ and the interface velocity $R'(t)$. To get such a relation, consider the continuity equation in spherical coordinates under the assumption of zero gradients along angular directions given by:

$$\frac{\rho_l}{r^2}\frac{\partial r^2 u_{l,r}}{\partial r} = 0 \quad \rightarrow r^2 u_{l,r} = constant \quad \rightarrow u_{l,r} = \frac{R^2}{r^2}u_{l,int} \tag{111}$$

Eq. (111) indicates that the liquid velocity has a maximum value at the interface and that the magnitude decreases with the radial location. The substitution of Eq. (111) in Eq. (82)

gives the relation between the interface velocity and the liquid velocity Eq. (112):

$$u_{l,r} = \frac{R^2}{r^2}\left(1 - \frac{\rho_v}{\rho_l}\right) R'(t) \tag{112}$$

The exact solution for the interface displacement and the liquid temperature can be found through a similarity variable [76]. Eq. (113) gives the interface displacement, Eq. (114) gives the temperature distribution on the liquid, Eq. (115) gives the constant β_{ls}, and Eq. (116) gives the similarity variable s:

$$R(t) = 2\beta_{ls}\sqrt{\alpha_l t} \tag{113}$$

$$T(s) = T_B - (T_B - T_{sat})\frac{\int_s^\infty s^{-2}\exp\left(-s^2 - 2\varepsilon{\beta_{ls}}^3 s^{-1}\right) ds}{\int_{\beta_{ls}}^\infty s^{-2}\exp\left(-s^2 - 2\varepsilon{\beta_{ls}}^3 s^{-1}\right) ds}, \quad r \geq R(t) \tag{114}$$

$$Ja = 2{\beta_{ls}}^3 \exp\left({\beta_{ls}}^2 + 2\varepsilon{\beta_{ls}}^2\right)\int_{\beta_{ls}}^\infty s^{-2}\exp\left(-s^2 - 2\varepsilon{\beta_{ls}}^3 s^{-1}\right) ds \tag{115}$$

$$s = \frac{r}{2\sqrt{\alpha_l t}}, \tag{116}$$

where $\varepsilon = 1 - \rho_v/\rho_l$. The only unknown parameter in Eq. (115) is the constant β_{ls}. A numerical method was used to find the constant β_{ls}. It is important to highlight that Eqs. (113)-(116) are equivalent to Scriven's solution. The only difference is that Eqs. (113)-(116) ignore dimensionless parameters and where rearranged to give more discernible expressions.

5 Results

The present section describes the results of the various simulated cases. Subsection 5.1 provides a description of the computational domain and operating conditions of the various analyzed cases. Subsection 5.2 discusses the accuracy and relevance of the proposed models to conduct the simulation of boiling. Subsections 5.3-5.5 show results of the simulation of planar interface evaporation, spherical bubble growth, and nucleate boiling.

5.1 Analyzed Cases

Figure 25(a) shows the computational domain and operating conditions in the simulation of planar interface evaporation. The boundary conditions are a wall with a constant temperature T_A at x =0, pressure outlet with a temperature T_B at $x = L$, and symmetric at $y = 0$ and $y = W$. The length of the domain L is such that the boundary has no influence on the temperature distribution near the interface. The width of the domain W consider only one computational cell. The initial conditions assumed an interface located at a certain position and phases with a temperature distribution. The theoretical equations described in section 4.1 evaluated at the time that corresponds to the interface position gave the initial temperature in the simulation. The reference properties for vapor and liquid phases are the following: $\rho_v = 0.001$ kg/m^3, $\rho_l = 0.001$ kg/m^3, $\mu_v = 0.01$ mPa-s, $\mu_l = 1$ mPa-s, $c_v = 200$ J/kg-K, $c_l = 200$ J/kg-K. Other phase change parameters include: $T_{sat} = 373$ K and $h_{fg} = 10,000$ J/kg. In addition, the cases with a density ratio higher than one were simulated by changing the liquid density while the vapor density remained constant and equal to $\rho_v = 0.001$ kg/m^3. Table 2 shows the magnitude of additional parameters used in the simulation. These properties were used unless stated otherwise. The properties were chosen based on simulations of boiling seen in the literature [9, 40, 78].

Figure 25(b) shows the computational domain and the operating conditions in the simulation of spherical bubble growth. The boundary conditions are axisymmetric at $r = 0$,

Table 2: Thermal conductivities and temperatures at the boundaries used in the simulation of Stefan problems.

Case	k_v (W/m-K)	k_l (W/m-K)	T_A (K)	T_B (K)
Sat-V/Sup-L	0.0005	0.005	373	383
Sup-V/Sub-L	0.005	0.0005	383	368
Sup-V/Sup-L	0.0005	0.005	378	383

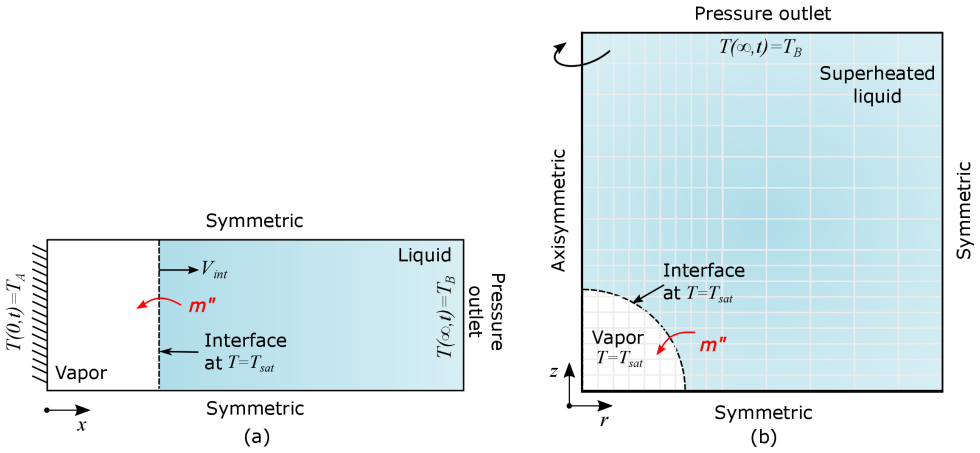

Figure 25: Computational domain and operating conditions: (a) planar interface evaporation, (b) spherical bubble growth.

pressure outlet with a temperature T_B at $r = L$ and at $z = L$, and symmetric at $z = 0$. The simulations consider an initial bubble radius $R_0 = 0.1$ mm. The initial patch had the reconstructed interface, meaning that the initial volume-fractions of the interface-cells correspond to the actual interface location and orientation. The theoretical equations of spherical bubble growth (Eqs. (114)-(116)) evaluated at the time t_0 gave the initial temperature distribution in the simulation. The time t_0 corresponds to the time required by the bubble to reach the radius R_0 as given by Eq. (113). User-Defined-Functions (UDFs) were used to import the initial temperature distribution into the computational domain (see section 3.3). The computational domain was a square of length L=0.5 mm. The simulation had a Courant number of less than 0.1, which led to a time step of 1x10^{-6} s for grid cells sizes of 1 μm. Simulation with a grid cell size of 0.2 μm had a time step of 1x10^{-7} s. The simulation of spherical bubble growth considered water properties at 1 atm with a superheat level of 5 K.

Figure 26 shows the computational domain and the operating conditions in the simulation of nucleate boiling. The boundary conditions are axisymmetric at $r = 0$, pressure outlet with a temperature T_{sat} at $z = L$, symmetric at $r = L$, and wall with a temperature T_w at $z = 0$. The simulations consider an initial bubble radius $R_0 = 0.12$ mm with a reconstructed interface. The initial temperature is a linear profile with $T = T_w$ at $z = 0$ and $T = T_{sat}$ at $z = 1$ mm. The length of the initial thermal layer comes from a correlation that gives the thermal boundary layer in turbulent natural convection [17]. However, the simulation ran for various bubble cycles to achieve a stable bubble growth pattern and thermal boundary layer. The operating conditions are a wall superheat level of 6.2 K. Table 3 provides other important conditions in the simulation of nucleate boiling. The simulation had a fixed Courant number of 0.25, which continuously adapted the time step based on the grid cell size and the interface velocity. The solution of the microlayer model (described in section 3.2) was used to account for the evaporative mass flux at the contact line.

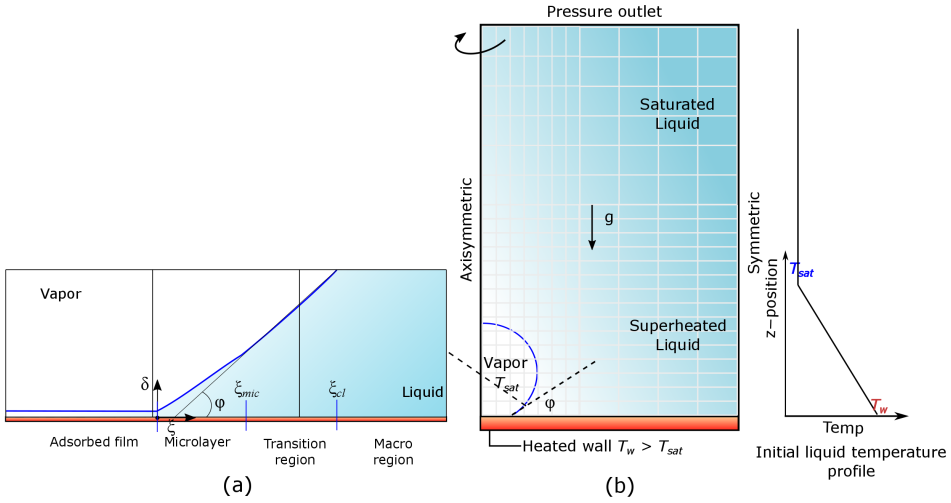

Figure 26: Simulation of nucleate boiling: (a) interface conditions near the contact line, (b) computational domain with boundary conditions, grid, and initial temperature profile.

Various recent simulation of nucleate boiling assume axisymmetric conditions [21, 33, 79, 80], which is a valid assumption since 3D simulations of nucleate boiling report a symmetrical thermal behavior [3, 12, 25]. Nevertheless, the extension of the proposed approach to 3D

Table 3: Parameters in the simulation of nucleate boiling.

Description	Symbol	Value
Contact angle	φ	40°
Domain length	L	5 mm
Domain width	W	3.5 mm
Initial bubble radius	R_0	0.12 mm
Saturation temperature	T_{sat}	373.15 K
Wall temperature	T_w	T_{sat} + 6.2 K
Fluid properties: water 1 atm		

simulations might be worth to attempt due to relevant information on bubble behavior showed by recent numerical studies [14,26,27]. Temperature drops of the order of 1 K or less are expected to occur at the contact line due to microlayer effects as reported in Kunklemann et al. [51]. However, simulations with a constant surface temperature show good agreement with experimental data on bubble growth [17].

5.2 Relevance and Accuracy of Proposed Methods

This section provides evidence of the accuracy and significance of the proposed models to perform the simulation of boiling (described in chapter 2). These methods include: (i) a one-cell method to interfacial temperature gradients, (ii) the use of an external software to compute interfacial temperature gradients, (iii) the interpolation of the interface-cells temperature with a linear approximation, and (iv) an interface-cells segregation algorithm to preserve the interface sharpness. In addition, this section discusses the effect of various surface tension models on the bubble shape.

5.2.1 Comparison of One-Cell Method Against Previous Methods

The evaluation of the performance of the proposed one-cell method (see section 2.1.2) considered a comparison against the models proposed by Ling et al. [7], Udaykumar and Shyy [2], and Sato and Niceno [3] (see section 1.2.1). The comparison evaluates the accuracy of each method on determining the temperature gradient needed for the evaporative mass flux

computation.

Ideal scenarios were recreated at three different locations along the interface of a spherical bubble. The radius of the spherical bubble is $R_0 = 0.1$ mm, and the scenarios lie at angles of inclination 5, 20 and 40° along the interface. The components of the scenario are the interface and nine cells. The nine cells represent the computational cells that would appear in the estimation of the temperature gradient in the simulation. The temperature of the cells is the theoretical temperature field, which was determined with Eq. (114) and with the radial position of the cell centers. The temperature field corresponds to a growing spherical bubble with a superheat level of 5 K and with fluid properties of water at 1 atm. The size of the computational cells is 1 μm.

Figure 27 shows the three different recreated scenarios for each method. The diagrams use gray color and lines to highlight the cells and probes needed for the estimation of the temperature gradient. Section 2.1 provided an explanation of the particular procedures on each method. It is important to observe that the methods of Ling et al. [7] and Udaykumar and Shyy [2] use the temperature of the interface-cell, which represents a disadvantage since the temperature of this cell requires an extra interpolation procedure. Sato and Niceno [3] proposed a method that estimates the temperature gradient by taking into account the interface temperature and position, but the temperature gradient is estimated at the neighboring-cell center rather than at the interface. Moreover, in some scenarios the method of Sato and Niceno should use interpolation procedures to find the temperature of the adjacent cells that have an interface (see for instance the scenario interface-2). The proposed one-cell method ignores the interface-cell and estimates the temperature gradient at the interface. In addition, the proposed method requires the control of fewer cells around the interface, which reduces the complexity of computing the evaporative mass flux.

Table 4 shows the results of the estimation of the temperature gradients with the various approaches. The theoretical temperature gradient is 1.58x10^6 K/m, which corresponds to the derivative of Eq. (114) evaluated at $r = R_0$.

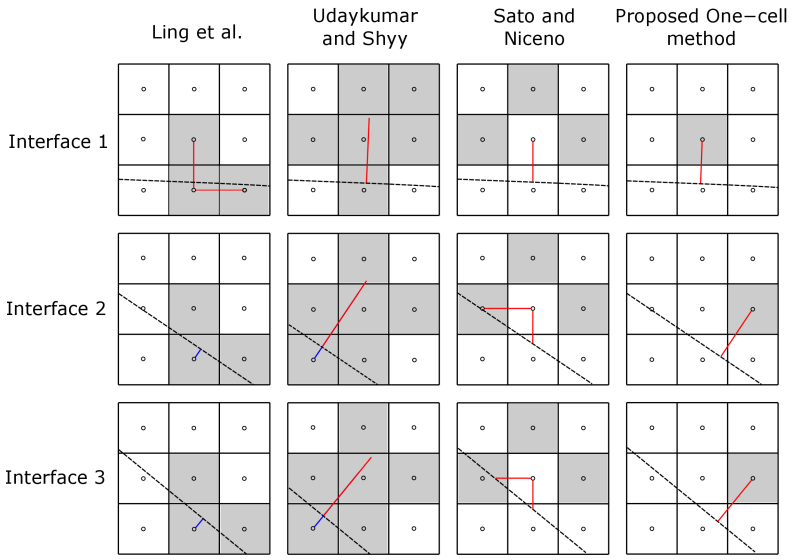

Figure 27: Diagrams for the estimation of ∇T of three different interfaces. Methods adopted for the evaluation: Ling et al. [7], Udaykumar and Shyy [2], Sato and Niceno [3], proposed one-cell method.

Table 4: Accuracy of analyzed methods to estimate the evaporative mass flux with various grid cell sizes. Bubble radius of 1 mm.

Method	**Interface-1**	**Interface-2**	**Interface-3**
Ling et al. [7]	0.92(38%)	1.48(1.98%)	1.31(13.4%)
Udaykumar and Shyy [2]	1.42(5.8%)	1.42(5.3%)	1.42(5.3%)
Sato and Niceno [3]	1.14(24.1%)	1.13(25.1%)	1.49(1.26%)
Proposed one-cell	1.48(1.79%)	1.47(2.5%)	1.44(4.17%)

The results in Table 4 indicate that the method of Udaykumar and Shyy [2] has a relative error in the range of 5.3% to 5.8%. We observed that a similar error was obtained if the temperature at the tip of the probe was estimated with the theoretical equation rather than the polynomial function. By using theoretical equations, we observed that the error was reduced to 1.2% if a second order backward approximation was adopted. Therefore, the 5% error is mainly due to the adoption of a first order backward approximation for the estimation of the temperature gradient. The method of Udaykumar and Shyy [2] can determine the interfacial temperature gradient with a second order approximation. However, this task requires the identification of more cells around the interface. Another important fact in Table 1 is that the method of Udaykumar and Shyy provides equivalent temperature

gradients with interface 1, 2 and 3. This is because the method determines the gradient with a probe of equal length on each interface analyzed.

The results in Table 4 indicate that the method of Ling et al. [7] has a relative error of 38%. Such a high error occurs because the method considers the temperature of cells in x and y directions, while the interface is almost horizontal. The authors proposed alternative procedures to reduce the error on these interfaces. The results also show that the method proposed by Ling et al. is minimal (1.98%) with interface-2, but it increases to 13.39% with interface-3. The reason for this trend is that the method estimates the temperature gradient at the interface-cell center rather than at the interface. Therefore, the minimal error is because interface-2 is closer to the interface-cell center relative to interface-3. The results indicate that method of Ling et al. [7] may lead to significant errors when the interface is far from this cell center.

The method proposed by Sato and Niceno [3] shows errors of the order of 20% with interface-1 and interface-2, and the error is minimal and equal to 1.26% with interface-3. The error is significant with interface-1 and interface-2 because these interfaces are far from the neighboring-cell center, which is the location at which the temperature gradient is estimated. The small error with interface-3 is due to the short distance between the interface and the neighboring-cell center. Also, the estimation assumes a second order approximation, which reduces the magnitude of the error.

The proposed one-cell method has an accuracy that is better than the multiple-cells method of Udaykumar and Shyy [2]. The multiple-cells method has a relative error in the range of 5.8% to 5.3% whereas the proposed one-cell method has a relative error is in the range of 1.79% to 4.17%. In addition, different to the methods of Ling et al. [7] and Sato and Niceno [3], the accuracy is maintained for all the analyzed interfaces with the proposed method. The error is only 1.79% with interface-1 and the error increases to 4.17% with interface-3. The variation is due to the change in length of the probe used for the estimation of the gradient. In the one-cell method, the length of the probe that finds the gradient is

Table 5: General terms of the various approaches for the computation of evaporative mass flux.

Method	**Relative error (%)***	**Location of ∇T estimation**	**Number of cells involved**	**Requires interpolation functions**
Ling et al.	2.0-38	Interface-cell center	3	Yes
Udaykumar and Shyy	5.3-5.8	Interface	6	Yes
Sato and Niceno	1.3-24.1	Neighboring-cell center	3	Yes
Proposed one-cell	1.7-4.2	Interface	1	No

given by the distance from the G-cell center to the interface, which is not always the same in all the interface-cells. However, it is important to emphasize that the simulation of bubble growth indicates that these small variations have no influence on the interface shape.

Table 5 presents a brief description of the main features of the various methods. The information in the table shows that the proposed one-cell method is more accurate than the methods in the technical literature. Only the proposed one-cell method and the method of Udaykumar and Shyy [2] estimate the temperature gradient at the interface, which helps to improve the accuracy of the mass transfer computation. Other methods assume a temperature gradient at a cell center. Another important fact is that the proposed one-cell method requires the identification of only one computational cell in the phase, which reduces the complexity of computing the evporative mass flux. Besides, the proposed method is free of interpolations functions since it ignores the interface-cell temperature. Moreover, the analysis showed that the proposed method applies for interfaces of different orientation and position. These characteristics provide strong evidence of the importance of the proposed method in simulations with interfacial mass flux.

5.2.2 Performance of One-Cell Method at Multiple Interface-cells

The evaluation of the one-cell method at various interface-cells along a spherical interface considered the Scriven's theory on spherical bubble growth [76]. We considered a compu-

tational domain with a spherical bubble immersed in superheated liquid (Scriven's theory provided the temperature of the liquid based on the bubble radius, the fluid properties, and the superheat level); the one-cell method computed the temperature gradients ($\frac{\partial T}{\partial n}_{num}$) at each interface-cell along the interface. The results obtained with the one-cell method were compared with the theoretical temperature gradient ($\frac{\partial T}{\partial n}_{th}$) (the derivative of Eq. (114) evaluated at the radial location of the interface).

The analysis considered two bubble radii of 0.1 mm and 1 mm. The 0.1 mm bubble had a computational domain with a length of 0.15 mm and the analyzed grid cell sizes were 1, 0.6, and 0.2 μm. The 1 mm bubble had a computational domain with a length of 1.5 mm and the analyzed grid cell sizes were 10, 6, and 2 μm.

Figure 28 shows the relative error of the temperature gradient on each G-cell along the interface. In the plot, θ is the angle measured from the vertical symmetrical axis along the clockwise direction. The plot shows results in a range from 0 to 45°, and similar results were observed in the range 45 to 90° with a symmetric line at θ =45°. The results correspond to a bubble of radius is 0.1 mm and to a grid cell size of 0.6 μm. The relative error is equal to $\left(\frac{\partial T}{\partial n}_{th} - \frac{\partial T}{\partial n}_{num}\right) / \frac{\partial T}{\partial n}_{num} \times 100$. The results show that the relative error follows an oscillatory trend along θ. The range of the relative error is 0.2 to 2.5%. From 0 to 5° the relative error increases and the distance between the G-cell center and the interface (length of probe-3 d_3) is proportional to θ. This is because the radial position of the G-cells remains constant, while the interface bends towards central axis. Around 5°, the relative error changes from 2.5% to 0.3% since the G-cell moved one cell below. A similar behavior occurs from 5 to 8°, the radial position of the G-cell remains constant, and the relative error augments because d_3 increases. These results show that the relative error is proportional to the length of probe-3. In the proposed approach, the maximum length of probe-3 is $d_3 = \sqrt{(1.5\Delta S)^2 + (1.5\Delta S)^2} d_3 = \sqrt{(1.5\Delta S)^2 + (1.5\Delta S)^2}$ (where ΔS is the grid cell size), which occurs when the interface is oriented at 45° and lies near the corner of the interface-cell. The results indicate that with a grid cell size of 0.6 μm, the maximum relative error is

2.5%; such small errors will not influence the bubble growth rate.

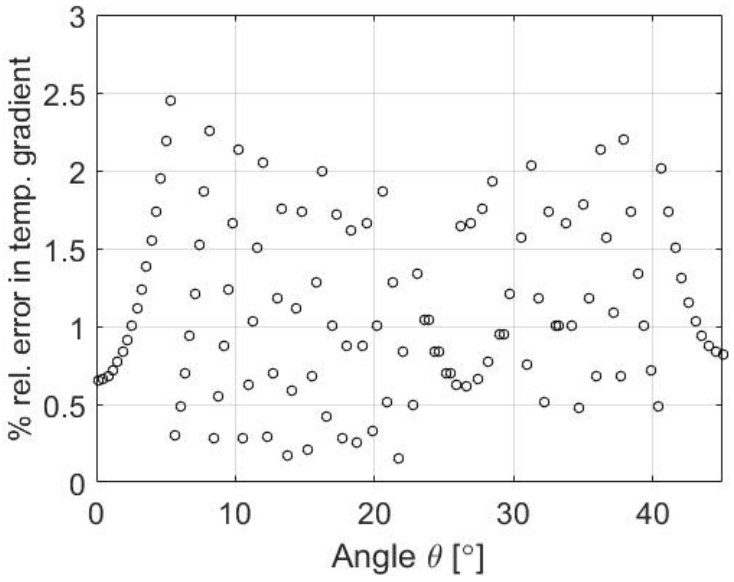

Figure 28: Accuracy on the estimation of interfacial temperature gradient along the normal direction with the proposed one-cell method. Bubble radius 0.1 mm, grid cell size 0.6 μm.

Figure 29 shows the average relative errors in the estimation of the interfacial temperature gradient. The analysis considered a bubble with a radius of 0.1 mm and with grid cell sizes of 1, 0.6, and 0.2 μm. The average relative error is $\sum_1^n rel.error_i/n$, where n is the total number of interface-cells. The results show a reduction in the average relative error as the grid is refined. The accuracy follows a quadratic trend despite the fact that the temperature gradient assumes a first order approximation. The deviations from the linear trend occur because the estimation of temperature gradient is along the normal direction and not along vertical or horizontal directions. As a result, the change in the grid cell size is not equivalent to the change in distance in the estimation of the temperature gradient. The non-linear dependence is stronger as the interface inclination gets closer to 45°. These results show that the proposed method takes more advantage of a reduction of the cell size relative to other approaches with gradients along Cartesian directions.

Another important result in Figure 29 is that small grid cell sizes lead to negative average relative errors. The negative value indicates that in some cells the numerical temperature

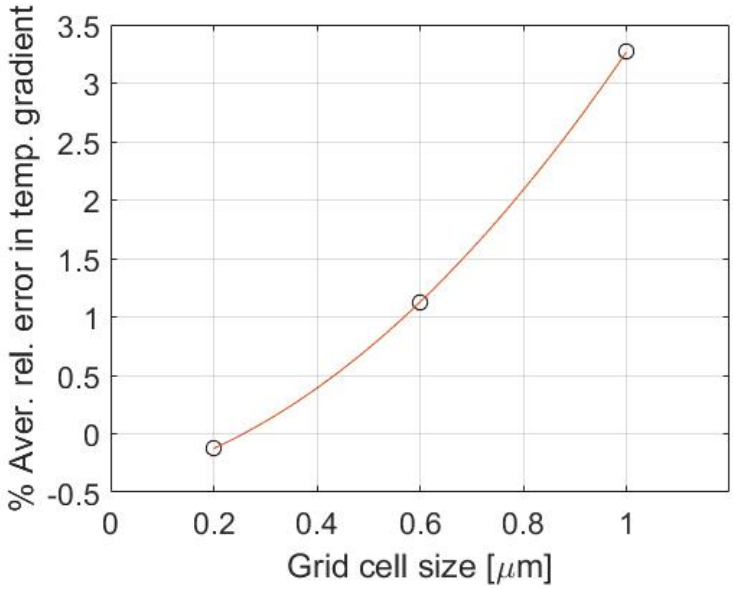

Figure 29: Accuracy of the proposed one-cell method to estimate interfacial temperature gradients with various grid cell sizes. Bubble radius equal to 0.1 mm.

gradient is greater than the theoretical temperature gradient. The relative error becomes negative when the distance between the G-cell and the interface d_3 is shorter than 0.25 μm. This occurs because the temperature gradient increases as the length d_3 becomes smaller. In fact, the limit of the temperature gradient as d_3 approaches zero is infinite. However, with the proposed method, the shortest distance is $\Delta S/2$ since the method locates the G-cell in the neighboring-cells (cells around the interface-cell).

Table 6 shows the results obtained for a bubble radius of 1 mm with grid cell sizes of 10, 6, and 2 μm. The small relative errors show that the proposed method works with bigger bubbles and with larger computational cells. The maximum relative error is small and the magnitude decreases with smaller grid cell sizes. The results indicate that bubble radii of 0.1 mm and 1 mm have equivalent relative errors. This occurs because in spherical bubble growth, the thickness of the thermal film is proportional to the size of the bubble. For example, the thickness of the thermal film is 12 μm with a bubble radius $R = 0.1$ mm, whereas the thickness of the thermal film is 120 μm with a bubble radius $R = 1$ mm.

Table 6: Accuracy of the proposed one-cell method to estimate evaporative mass flux with various grid cell sizes. Bubble radius of 1 mm.

Grid cell size (μm)	Min. % rel. error	Max. % rel. error	Aver.% rel. error
10	0.75	6.33	3.27
6	0.21	2.45	1.13
2	-0.33	0.11	-0.12

5.2.3 Performance of External-software Mass Transfer Model

The proposed external-software method uses MATLAB to interpolate the temperature at a reference point in the estimation of interfacial temperature gradients (see section 2.1.3). The present subsection compares the results of such a proposed method against theoretical estimations.

Consider a semicircular bubble over a heated surface where the liquid has a linear temperature profile; the semicircular bubble has a radius R_0 and forms a contact angle φ with the surface (see Figure 26). Under these conditions, it is easy to derive an equation that provides the temperature at a certain distance (reference point) from the interface. The reference temperature is $T_{ref} = T_w - (\Delta T_w/\delta_{th})\, z_{ref}$, where z_{ref} is the z-coordinate of the reference point. The theoretical temperature gradient with a first order approximation is $\frac{\partial T}{\partial n}_{th} = (T_{ref} - T_{sat}) \,/\, (R_{ref} - R_0)$, where R_0 is the radius of the circular bubble and R_{ref} is the radius of the reference points near the interface. The coordinate z_{ref} relates with R_{ref} by $z_{ref} = \pm R_{ref}\sqrt{1/\left(1 + tan\theta^2\right)} + h$, where θ is the angle of the reference point (measured from the vertical boundary in the clockwise direction), and h is the z-coordinate of the center of the circular bubble.

Figure 30 compares the numerical and the theoretical interfacial temperature gradients for three different grid cell-sizes of 20, 16, and 12 μm and two different order of approximations. Figure 30(a) shows temperature gradients obtained with a first order approximation Eq. (13), and Figure 30(b) shows temperature gradients obtained with a second order approximation Eq. (14). The bubble radius is 0.12 mm, the contact angle φ is 40°, and the temperature is linear with a thermal thickness of 1 mm. In the graphs, θ is the angle of the

reference point measured from the vertical boundary in the clockwise direction. Due to the linear temperature profile, the bubble top ($\theta = 0°$) has the minimum temperature gradient and the bubble contact-line (θ =140°) has the maximum temperature gradient. Results in Figure 30(a) and (b) indicate excellent agreement between the proposed approach to estimate $\frac{\partial T}{\partial n}_{int,l}$ (using interpolation functions with an external software) and the theoretical estimations. The good agreement proves the effectiveness of the proposed approach to compute the interfacial gradients and the correct estimation of the reference points using normal vectors. However, small errors appear near the contact line with a large grid cell size of 20 μm due to inaccuracies of the normal vector as the grid cell size increases. A comparison between Figure 30(a) and Figure 30(b) shows that the order of approximation significantly influences the magnitude of the interfacial temperature gradients along the normal direction. A first order approximation gives smaller temperature gradients relative to second order approximation for the same grid cell size. In addition, the difference between two different grid cell sizes is larger with a second order approximation. These results reveal that both order of approximations lead to the exact value, but the second order approximation gets faster to the theoretical solution. In the simulation of bubble growth, the temperature gradients might be smaller due to the interface-liquid interaction and fluid behavior.

5.2.4 Performance of Model to Include Sharp Interface with Tsat

In the present work, the simulations of spherical bubble growth and nucleate boiling interpolate the temperature of the interface-cells to include the interface saturation temperature. The present work proposed to interpolate the temperature of the interface-cells with a linear temperature profile approximation near the interface and with interfacial temperature gradients (see section 2.2).

To analyze the accuracy of the proposed approach on determining the temperature of the interface-cell, we considered a computational domain with a spherical bubble of $R_0 = 0.1$ mm. Scriven's theory gave the temperature in the computational domain based on the bub-

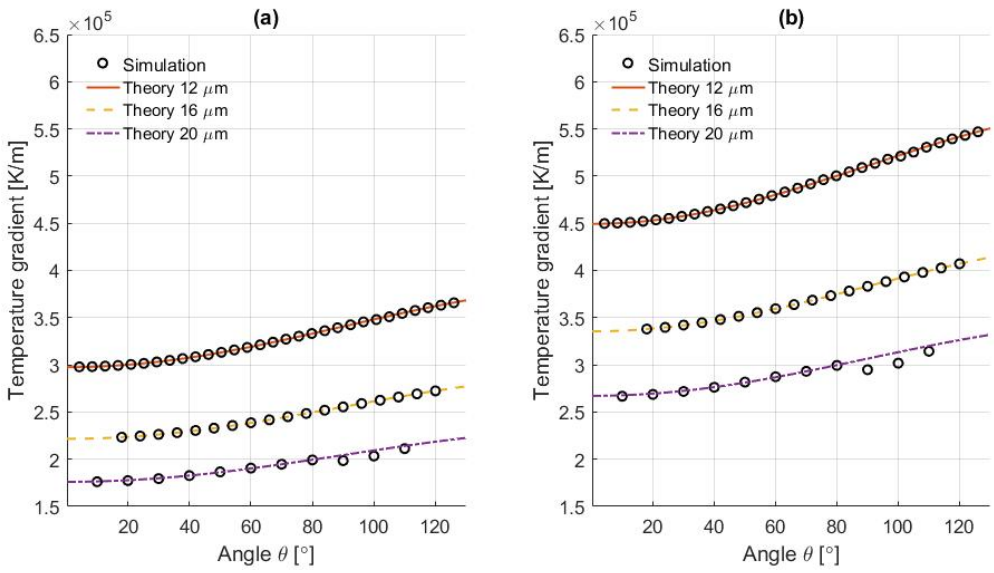

Figure 30: Comparison between theory and proposed model for the estimation of interfacial gradients along the normal direction with the proposed external-software method. (a) First order approximation, and (b) second order approximation.

ble radius, the fluid properties and the wall superheat level. The proposed Eq. (18) gave the numerical temperatures of the interface-cells, and Scriven's solution Eq. (114) determined the theoretical temperatures of the interface-cells. Scriven's solution is valid only for the liquid side, but Eq. (114) has a continuous temperature variation through the interface. Therefore, the theoretical solution provides interface-cell temperatures that preserve a uniform temperature gradient through the interface. $\left(T_{IC,th} - T_{IC,num}\right)/T_{IC,th} \times 100$ gave the percentage of relative error.

Figure 31(a) shows the relative error of the temperature on each interface-cell along the interface with a grid cell size is 0.6 μm. The results show a relative error in the range -0.1 to 0.17%. In addition, the relative error changes from positive to negative values as the interface moves through the interface-cell. For example, in the range 0 to 2.5° the relative error decreases from 0.05 to 0%, and in the range 2.5 to 5°, the relative error becomes negative. At 5° the error varies from -0.1 to 0.12%, which is due to a change in the radial location of the G-cell; this jump increases the distance between the G-cell and the interface d_3. These results show that the relative error decreases as the interface gets closer to the interface-cell center,

and the relative error is zero when the interface is in contact with the interface-cell center. Figure 31(b) shows the temperature of the interface-cells along the interface. The results indicate that the temperature of the interface-cells (T_{IC}) oscillates around the saturation temperature T_{sat} =373.15 K. The magnitude of the interface-cell temperature depends on the position of the interface: $T_{IC} < T_{sat}$ when the interface lies above the interface-cell center, and $T_{IC} > T_{sat}$ when the interface lies below the interface-cell center. These results show that the estimated value for the interface-cell temperature preserves the temperature profile near the interface, which helps to compute accurate temperature gradients at the faces of the interface-cells.

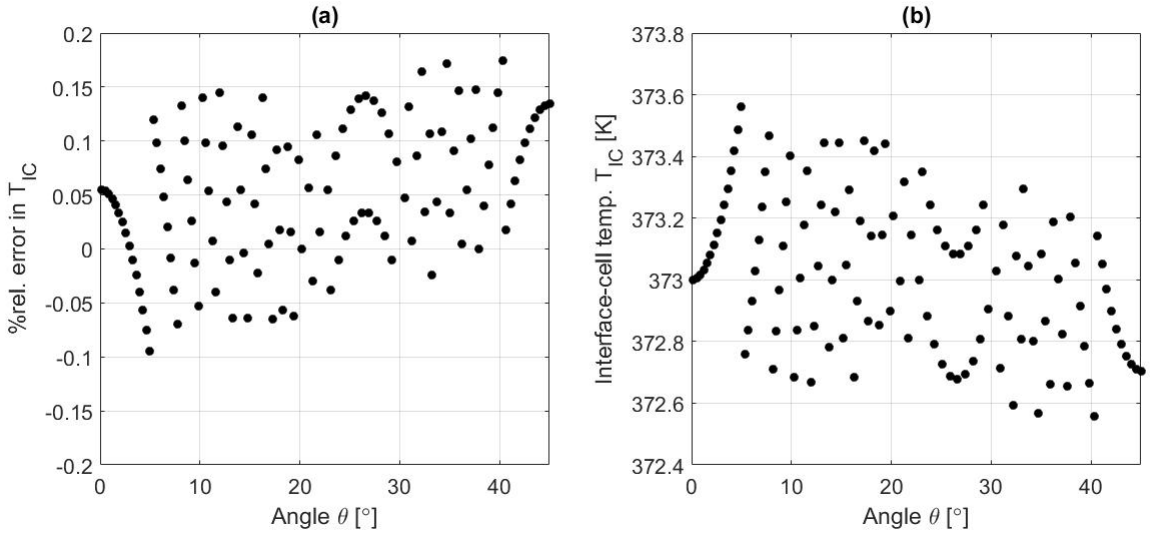

Figure 31: Accuracy on the estimation of the interface-cell temperature T_{IC} along the interface. (a) Percentage of relative error (b) magnitude of interface-cell temperature. Bubble radius equal to 0.1 mm.

Table 7 shows the relative errors of the estimated interface-cells temperatures for grid cell sizes of 1, 0.6, and 0.2 μm. Results show maximum relative errors of 0.25%; this error can reduce to 0.07% by augmenting the grid cell resolution. The average relative error is smaller than 0.08%, which shows that the proposed method accurately determines the temperature of the interface-cells. The dependence between the grid cell size and the average relative error follows a linear trend. This trend occurs because the method performs a linear interpolation along the direction normal to the interface to find the temperature of the interface-cell.

Table 7: Accuracy of the proposed method to estimate interface-cell temperature with various grid cell sizes. Bubble radius of 0.1 mm

Grid cell size (μm)	Min. % rel. error	Max. % rel. error	Aver.% rel. error
1	-0.14	0.25	0.077
0.6	-0.09	0.17	0.051
0.2	-0.01	0.07	0.028

5.2.5 Effect of Model to Include Sharp Interface with Tsat

This section shows the effect of the proposed model that includes the interface saturation temperature in the simulation of spherical bubble growth and nucleate boiling (see section 2.2.2). The simulation interpolates the temperature of the interface-cells to include the interface temperature (ghost fluid method). The proposed model interpolates the temperature of the interface-cells with a linear temperature profile approximation. The study considered a constant mass flux at the interface to prevent the influence of the temperature field near the interface on the bubble growth process. The magnitude of the mass evaporative flux is $m_e'' = 0.1$ kg/s-m^2, which corresponds to an average temperature gradient of 4x10^5 K/m. The analysis considered two cases: case (a) adopts the energy equation given by ANSYS-Fluent, which assumes interface-cells with average properties, and case (b) interpolates the temperature of the interface-cells to consider the interface location and temperature. Both cases consider liquid with initial linear temperature profile and saturated vapor.

Figure 32(a) shows the temperature obtained with case (a) and Figure 32(b) shows the temperature obtained with case (b). In Figure 32(a), both sides of the contour correspond to the same data; however, the left side displays a temperature range of 350-378.15 K and the right side displays a temperature range of 373.15-378.15 K (cells with a temperature outside the range have a transparent color). The left side of Figure 32(a) shows that simulations with average properties have temperatures of the order of 350 K near the contact line region on the liquid side. The right side of Figure 32(a) reveals that simulations with average properties lead to a disconnection between the liquid temperature and the interface conditions (the liquid is unable to generate the thermal film since the simulation ignores the interface

temperature). Figure 32(b) indicates that the interpolation of the interface-cells temperature with the proposed model generates temperatures within the expected range near the contact line. Moreover, the simulation with the proposed model shows that the liquid changes its temperature based on the interface conditions (the liquid temperature changes to reach the saturation temperature at the interface).

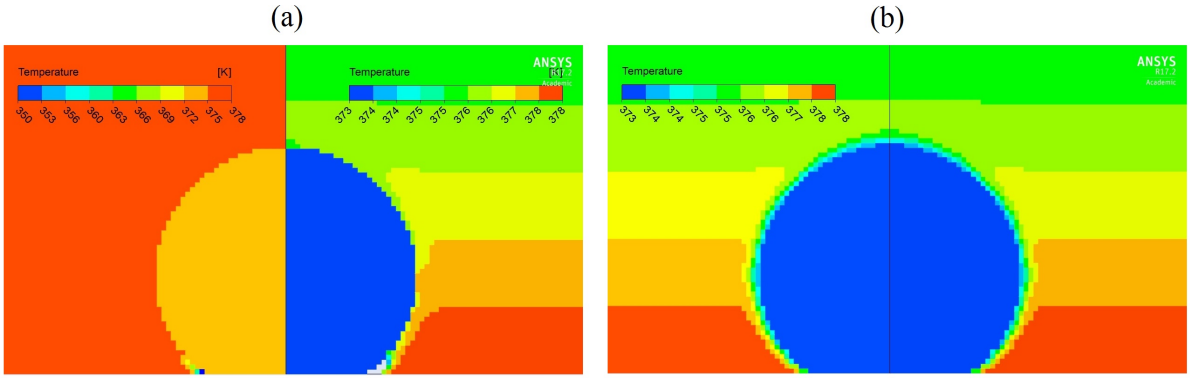

Figure 32: Temperature distribution with two approaches: (a) traditional methods at interface-cells (two different ranges of temperature), (b) sharp interface model. Simulation time $t = 0.12$ ms.

5.2.6 Effect of Segregation Algorithm to Keep Interface Sharpness

This section shows the benefits of the proposed model to avoid the interface deformation while declaring mass transfer (see section 2.3). The proposed interface-sharpness algorithm classifies interface-cells in two different types: mass-transfer-cells and unsuitable-interface-cells. Mass-transfer-cells are interface-cells that lie next to vapor cells and that have mass transfer source terms. Unsuitable-interface-cells are interface-cells that lie next to mass-transfer cells and that have zero mass transfer.

The evaluation of the relevance of the proposed interface-cells segregation algorithm considered a bubble growing over a heated surface with a constant mass flux. In bubble growth with a constant evaporative mass flux ($m_e'' = 0.1$ kg/s-m^2), the temperature field plays no role on the bubble growth process, which eliminates the influence of the mass and interface-cells interpolation models in the simulation. The study considers two different

cases: case (a) is a simulation that uses the proposed segregation algorithm to preserve the sharp interface, and case (b) is a simulation that declares mass transfer in all the cells with an interface.

Figure 33 shows the identification of the mass-transfer-cells and neighboring-mass-transfer-cells in the simulation. Figure 33(a) makes a zoom at the bubble-edge where $20° \leq \theta \leq 45°$, and Figure 33(b) makes a zoom at the bubble where $70° \leq \theta \leq 105°$ (θ is the angle measured from the vertical axisymmetric boundary in the clockwise direction). The figure reports values of the User-Defined-Memory-3 (UDM3) on computational cells used to classify the interface-cells. Cells with UDM3 = 0 have blue color and correspond to vapor or liquid cells, cells with UDM3 = 1 have green color and correspond to mass-transfer-cells (interface-cells with mass transfer), cells with UDM3 = 2 have red color and correspond to neighboring-mass-transfer-cells (cells located next to mass-transfer-cells that might have an interface). The circular line crossing the computational cells shows the interface location. The results in Figure 33 indicate that the proposed segregation algorithm identifies the interface-cells that lie next to vapor-cells as mass-transfer-cells (computational cells with UDM3 = 1). In addition, the algorithm identifies the computational cells that lie next to mass-transfer-cells as neighboring-mass-transfer-cells (computational cells with UDM3 = 2). Results show interface-cells that are neighboring mass-transfer cells; the segregation algorithm classifies these cells as unsuitable mass transfer cells. If the interface lies within a neighboring-mass-transfer cell, the algorithm adds such an interface length to the corresponding mass-transfer cell. The proposed algorithm declares mass transfer only on interface-cells that lie next to vapor cells to preserve the interface sharpness.

Figure 34 shows the liquid volume-fractions near the contact line region at 20 μs (the maximum z and r coordinates are 50 μm and 150 μm, respectively). Figure 34(a) shows the results of a simulation that declares mass transfer in all the cells that have an interface (interface-cells) and Figure 34(b) shows the results of a simulation that declares mass transfer only in interface-cells that lie next to vapor cells (mass-transfer-cells). Results in Figure 34(a)

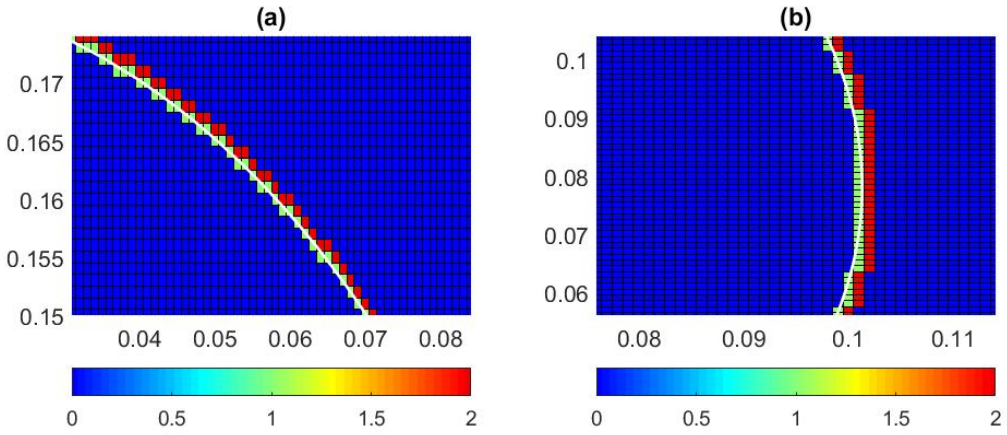

Figure 33: Contours of UDM3; UDM3 = 1 identifies mass-interface-cells, UDM3 = 2 identifies neighboring-mass-transfer cells. Bubble-edge at: (a)$10° \leq \theta \leq 25°$, (b) $40° \leq \theta \leq 60°$. Axis dimensions in mm.

indicate that the declaration of mass transfer in all the interface-cells deforms the interface. Such a simulation fills all the interface-cells at a similar rate, which diffuses the liquid volume-fractions. The interface deformations lead to inaccurate normal vectors, wrong interfacial temperature gradients, and incorrect surface tension effects. Results in Figure 34(b) indicate that that the declaration of mass transfer only in mass-transfer-cells maintains the interface sharpness. The proposed method eliminates the numerical diffusion of the volume-fractions and defines a sharp and well-defined interface. The segregation algorithm keeps a sharp interface by putting order in the filling of the interface-cells; the method gives priority to interface-cells next to vapor-cells for the transfer of mass, which prevents the deformation of the interface.

5.2.7 Surface Tension Effects

The simulation uses the Green-Gauss theorem to compute the gradients of any scalar at the cells center. The present work considers three different cases to estimate the volume-fraction gradients in the computation of surface tension effects: (a) node-based gradients, (b) cell-based gradients method, and (c) node-based gradients of smoothed volume-fractions. Case (a) considers the arithmetic average of the nodal values on the face, case (b) considers

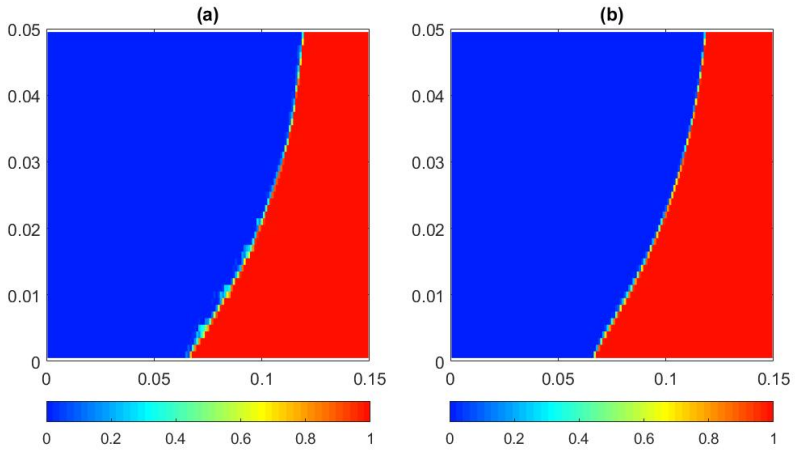

Figure 34: Bubble shape near the contact line region at 20 μs. (a) Simulation that declares mass transfer in all the interface-cells. (b) Simulation that uses the proposed algorithm to identify mass-transfer-cells. Axis dimensions in mm.

the arithmetic average of the values at the neighboring-cell centers, and case (c) considers smoothed volume-fractions to estimated nodal values. The analysis of the surface tension computation considered a bubble growing over a surface with a constant evaporative mass flux. A constant mass flux eliminates possible errors associated with the estimation of mass transfer with interfacial temperature gradients. The simulation adopts the proposed algorithm to identify the mass-transfer-cells and to include the sharp interface. The initial conditions consider a bubble of radius 0.12 mm.

Figure 35 shows liquid volume-fractions at 80 μs for the three analyzed methods to compute surface tension. Case (a) show appreciable interface deformations around the lateral sides of the bubble and near the contact line. The interface smears over three to four computational cells and the smearing increases with time. Case (b) generates no interface deformations; however, the bubble shape changes to a non-circular shape with flat lateral sides. Case (c) creates a bubble-edge without deformations; in addition, the bubble shape remains circular. These results reveal that surface tension computation plays an important role on preserving the interface-sharpness and bubble edge. Simulations of boiling with VOF should compute surface tension with smeared volume-fractions to preserve the interface

sharpness and the bubble shape. Welch and Wilson [40] simulated film boiling by using VOF with a sharp interface and by smoothing the volume-fractions to compute surface tension. However, the approach was never tested for the simulation of nucleate boiling. To our knowledge, this is first report that simulates nucleate boiling with a sharp interface, the proposed interface-cells segregation model eliminates the interface deformations near the contact line region (see section 5.2.6).

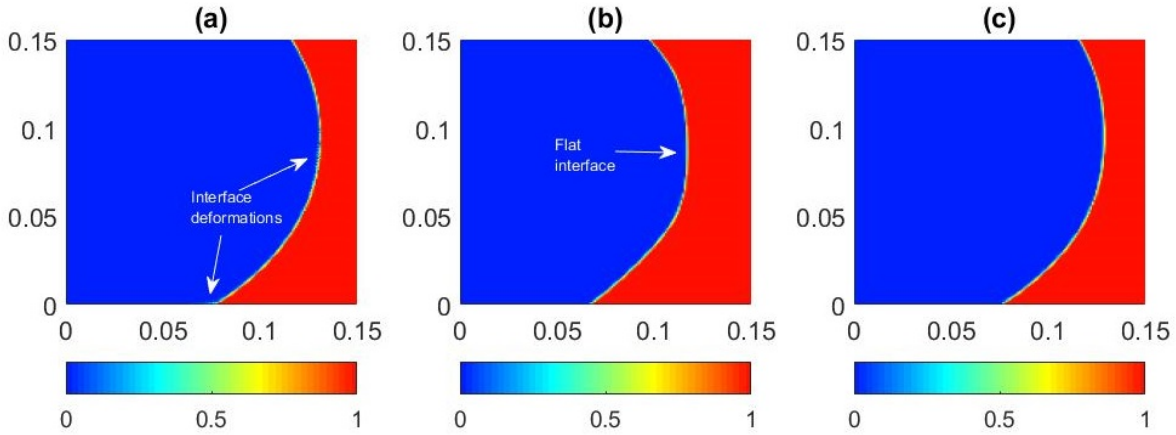

Figure 35: Effect of surface tension on liquid volume-fractions of a bubble growing over a surface with constant evaporative mass flux: (a) node-based gradients, (b) cell-based gradients, and (c) node-based gradients with smoothed volume-fractions. Axis dimensions in mm.

5.3 Simulation of Planar Interface Evaporation

To validate film boiling simulations, previous works have used theoretical solutions of one-phase Stefan problem with superheated vapor and saturated liquid to validated their results [9, 16, 65]. To validate pool boiling simulations, previous works have used semi-theoretical solution of one-phase Stefan problem with saturated vapor and superheated liquid with density effects (sucking problem) [3, 10, 40]. The present work proposes fully theoretical solutions for sucking problem and for other proposed cases with heat transfer on both liquid and vapor phases (two-phase Stefan problems), see section 4.1. The present section compares the numerical results against the theoretical solutions to prove the validity of the simulation

and theoretical equations. Section 5.1 described the computational domain and operating conditions used to simulate planar interface evaporation. Simulations of planar interface evaporation consider the sharp interface model described in section 2.2.1, which includes the interface as a boundary in the discretization of the neighboring-cells. The result in the following sections use lines and circles to represent the numerical and theoretical data, respectively.

5.3.1 Simulation of One-Phase Stefan Problems

In Stefan problems with superheated vapor and saturated liquid that consider or neglect density effects (Sup-V/Sat-L and Sup-V/Sat-L-ρ), the liquid properties have no effect on the interface displacement and temperature distribution since the liquid is remains at a constant temperature (saturation temperature). Therefore, Sup-V/Sat-L and Sup-V/Sat-L-ρ have the same theoretical and numerical solution. Figure 36 shows results of one-phase Stefan problems with Sup-V/Sat-L and Sup-V/Sat-L-ρ. The figure compares the numerical interface displacements against the theoretical equation Eq. (66). The results show excellent agreement between the theoretical and numerical results for coarse, medium, and fine grid cell sizes. At 1 s, the relative error of interface position is 0.01 % with 2000 μm grid cell size (coarse grid). The good agreement between the theoretical and numerical data indicates that the proposed sharp interface model is an effective method to simulate evaporation at planar interfaces.

Figure 37 shows the numerical and theoretical results for Stefan problems with saturated vapor and superheated liquid that consider or neglect density effects (Sat-V/Sup-L and Sat-V/Sup-L-ρ). Figure 37(a) shows the temperature distribution at 1 s, and Figure 37(b) shows the interface displacement. In the figures, ρ_r represents the density ratio $\rho_r = \rho_l/\rho_v$, where the vapor density remains constant and equal to ρ_v=0.001 kg/m^3. The simulation considered different initial interface positions for each density ratio; the number of computational cells inside the thermal film defined the initial interface position (a minimum number of 20

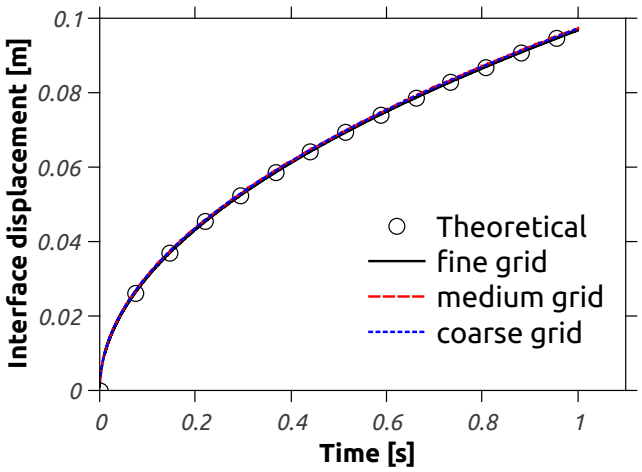

Figure 36: Interface displacement in one-phase Stefan problem with Sup-V/Sat-L and Sup-V/Sat-L$-\rho$.

cells). The simulation considered grid cell size of 400 μm. Figure 37(a) shows that vapor remains with a constant temperature of 373.15 K, whereas the liquid changes its temperature from 373.15 K at the interface to 378.15 K at a distance far from the interface. The liquid temperature forms a thermal film near the interface, which thickness depends on the density ratio. The thickness of the thermal layer on the liquid side decreases with higher density ratios (higher liquid densities). Higher liquid densities imply lower thermal diffusivities ($\alpha_l = k_l/(\rho_l c_{pl})$); therefore, a higher liquid density decreases the diffusion of heat on the liquid, which leads to thinner thermal films. In addition, results reveal that excellent agreement between the numerical and theoretical temperatures, which proves the validity of both approaches to predict the temperature distribution. Figure 37(b) indicates that the interface moves faster with liquids that have a higher density. The lower heat diffusion on liquids with higher densities leads to thinner thermal films and large temperature gradients at the interface, which increases the interface displacement. Results show excellent agreement between the numerical and the theoretical interface displacements for the various analyzed density ratios.

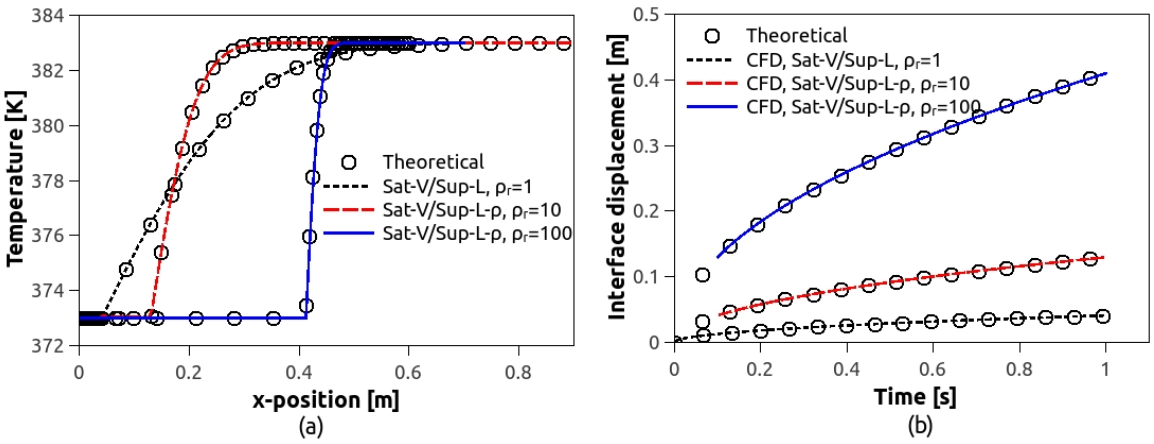

Figure 37: Comparative results for Stefan problem with Sat-V/Sup-L ($\rho_r = 1$) and Sat-V/Sup-L-ρ ($\rho_r = 10$, 100): (a) temperature distribution at 1 s, (b) interface location at different times. $\rho_r = \rho_l/\rho_v$.

5.3.2 Simulation of Two-Phase Stefan Problems

Two-phase Stefan problems consider heat transfer on the vapor and liquid phases. To our knowledge, this is the first work that considers two-phase Stefan problems to validate simulations of boiling. The two-phase Stefan problems consider four different cases: (i) superheated vapor and subcoooled liquid (Sup-V/Sub-L), (ii) superheated vapor and superheated liquid (Sup-V/Sup-L), (iii) superheated vapor and subcooled liquid with density effect (Sup-V/Sub-L-ρ), and (iv) superheated vapor and superheated liquid (Sup-V/Sup-L) with density effect (Sup-V/Sup-L-ρ). As explained in section in section 4.1.3, a density difference between phase introduces a convective term in the governing equations. The convective term appears because the vapor expands, which pushes the liquid outwards.

Figure 38 shows the numerical and theoretical results for a Stefan problems with superheated vapor and subcooled liquid that consider or neglect density effects (Sup-V/Sub-L and Sup-V/Sub-L-ρ). The figure includes the case of Sup-V/Sat-L for comparison purposes. Figure 38(a) shows the temperature distribution at 1 s, whereas figure 38(b) shows the interface displacement. Cases with a density difference consider a density ratio of ten. This low density ratio avoids the formation of thin thermal layers that require small grids as shown in

the previous section. The simulations used a grid cell size of 400 μm. Also, the simulations of Stefan problem with a density difference were initialized at 0.1 s to avoid thin thermal layers that appear when the time is near zero seconds (simulations with thin thermal layers require smaller grid cell sizes and time steps, which increases the computational time significantly). In addition, the case with a higher density ratio has a higher liquid thermal conductivity (k_l =0.005 W/m-K) to avoid thin thermal layers.

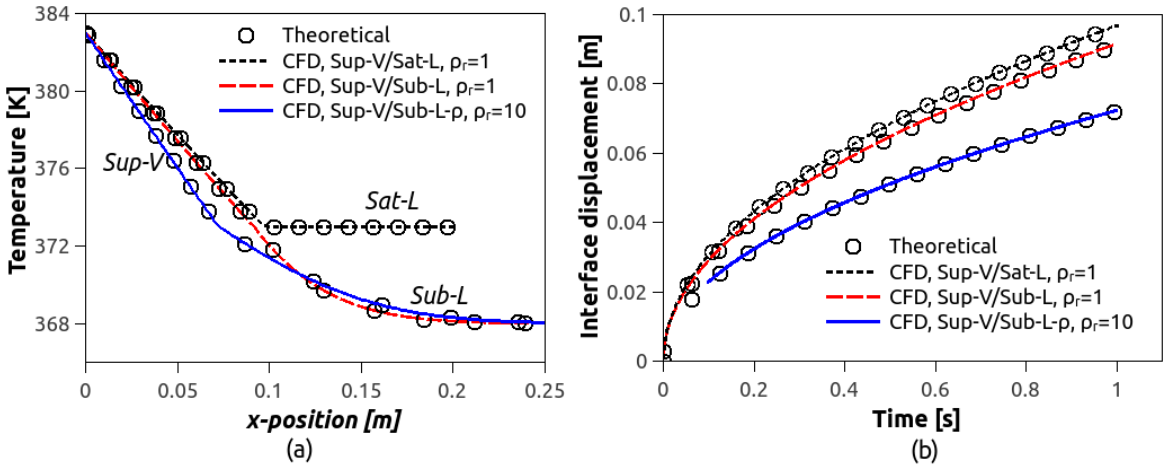

Figure 38: Comparative results for Stefan problem with Sup-V/Sat-L, Sup-V/Sub-L (ρ_r = 1), and Sup-V/Sub-L-ρ ($\rho_r = 10$) (a) temperature distribution at 1 s, (b) interface location at different times. $\rho_r = \rho_l/\rho_v$.

Figure 38(a) shows that the superheated vapor changes its temperature from 383.15 K at $x = 0$ to 373.15 K at the interface ($x = X(t)$), whereas the subcooled liquid changes its temperature from 373.15 K at $x = X(t)$ to 368.15 K at $x = \infty$, where ∞ is a large distance from the interface; saturated liquid keeps its temperature at 373.15 K. The case of Sup-V/Sub-L-ρ should show a temperature profile with a thinner thermal layer on the liquid side since a higher liquid density decreases the liquid thermal diffusivity. However, this effect is not observed Sup-V/Sub-L-ρ has a higher thermal conductivity (k_l =0.005 W/m-K) relative to Sup-V/Sub-L (k_l =0.0005 W/m-K). Figure 38(b) exhibits the effect of a subcooled phase on the interface displacement. As expected, the interface moves faster if the liquid is saturated because all the heat arriving at the interface is used for evaporation. However, if

the liquid is subcooled, the interface moves slower due to the heat lost from the interface to the liquid, which implies less heat available for evaporation. The results also show that cases where a higher liquid density and thermal conductivity decreases the interface velocity. This is because the sensible heat from the interface to the liquid increases due to its higher thermal conductivity.

Figure 39 shows the numerical and theoretical results for a Stefan problems with superheated vapor and superheated liquid that consider or neglect density effects (Sup-V/Sup-L and Sup-V/Sup-L-ρ). The figure includes the case of Sat-V/Sup-L for comparison purposes. Figure 39(a) shows the temperature distribution at 1 s, whereas figure 39(b) shows the interface displacement. The results in Figure 39(a) indicate that the simulation accurately reproduces the theoretical data despite the peak in temperature at the interface. These results illustrate the effectiveness of the present method to deal with abrupt changes in temperature at the interface. The change in temperature is properly captured because the integration of the terms in the governing equations over the neighboring-cells considers both the exact interface position and temperature throughout the phase change process.

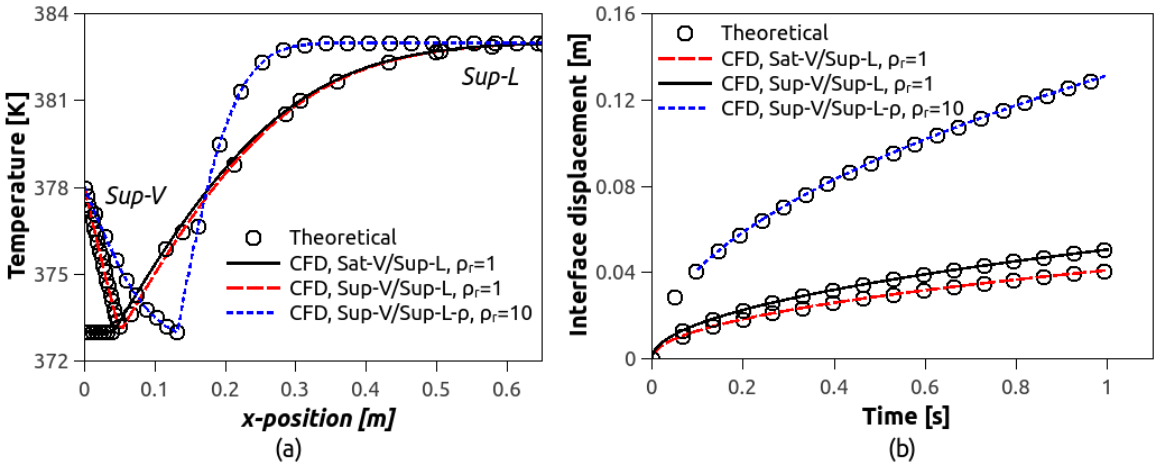

Figure 39: Comparative results for Stefan problem with Sat-V/Sup-L, Sup-V/Sup-L ($\rho_r = 1$), and Sup-V/Sup-L-ρ ($\rho_r = 10$) (a) temperature distribution at 1 s, (b) interface location at different times. $\rho_r = \rho_l/\rho_v$.

The temperature profiles in Figure 39(a) indicate that the cases Sat-V/Sup-L and Sup-

V/Sup-L have similar liquid temperature distributions. These results imply that the heat transfer of the vapor has little influence on the heat transfer on the liquid. The case of Sup-V/Sup-L-ρ has a thinner thermal layer since the higher liquid density decrease the thermal diffusivity. Another important observation is that the vapor temperature of Sup-V/Sup-L-ρ has a non-linear profile with low temperature gradients at the interface, which may be due to the faster interface velocity. The vapor with low thermal conductivity is unable to transfer enough heat to the vapor being generated during the evaporation and, therefore, the vapor temperature gradient at the interface is low. Figure 39(b) shows the interface displacement for the two-phase Stefan problems with two superheated phases. A comparison between Sat-V/Sup-L and Sup-V/Sup-L indicates minimal increase in the interface velocity, which shows that the contribution of the superheated vapor to the mass transfer is insignificant. The fact that Sup-V/Sat-L-ρ has a much faster interface velocity indicates that the heat flux to the interface increases significantly at higher liquid densities. This is because the higher liquid density leads to smaller heat diffusion rates and higher liquid temperature gradients. The larger liquid temperature gradients lead to higher mass transfer rates, which increases the interface displacements.

5.4 Simulation of Spherical Bubble Growth

Spherical bubble growth is a more complex simulation that involves geometrical calculations to account for the interface effects. The analysis considers adiabatic bubble growth and spherical bubble growth. Adiabatic bubble growth has a constant evaporative mass flux at the interface; a comparison with theory ascertains a proper solution of the momentum and volume-of-fluid equations. Non-adiabatic bubble growth has an evaporative mass flux that depends on the temperature gradient at the interface; a comparison with theory ascertains a proper solution of the energy equation, and a proper estimation of the mass transfer based on the interface position and orientation.

5.4.1 Adiabatic Bubble Growth

The simulation of adiabatic bubble growth had an initial bubble of radius 0.1 mm. The simulation used the proposed segregation algorithm (see section 2.3) to keep the interface sharp. The segregation algorithm prescribed mass transfer only at mass-transfer-cells. At each time step, UDFs identify the mass-transfer-cells and unsuitable-mass-transfer-cells and determine the mass source term in Eq. (22) based on the constant evaporative mass flux. It is important to emphasize that in the present work, the interface was not smeared and the mass source terms were not distributed into two to three cells as previously done in several works. The distribution of the mass source terms is usually done to preserve the stability of the simulation. However, the simulations in the present work declare mass transfer only on cells that have an interface. The imposed mass flux is the average mass flux in a non-adiabatic bubble in a time interval. The time interval corresponds to 1 ms with an initial bubble of radius 0.1 mm. The magnitude of the imposed mass flux is m_e''=0.18 kg/s-m^2. The simulation considered two different grids cell sizes of 10 and 1 μm.

Figure 40 compares the numerical and the theoretical bubble radius. A comparison between numerical and theoretical bubble radius points out issues in mass conservation and interface shape during bubble growth. The results show excellent agreement between the numerical and the theoretical bubble radius as a function of time. In addition, both grid cell sizes provide a similar level of mass conservation. The good agreement is because by nature the VOF method is mass conservative since it moves the interface with changes in volume. Moreover, in this problem the generation of vapor drives the movement of the interface. The advection effect dictated by the vapor velocity in the VOF equation is negligible since the vapor remains static during the phase change process. The good agreement between theory and simulation shows evidence of an accurate interface tracking when there is mass transfer between two phases divided by a sharp interface.

Figure 41 shows the liquid volume-fractions F_l at 0.5 ms for grid cell sizes of 10 and 1 μm. The numbers around the contour give the coordinates in millimeters. The vapor and liquid

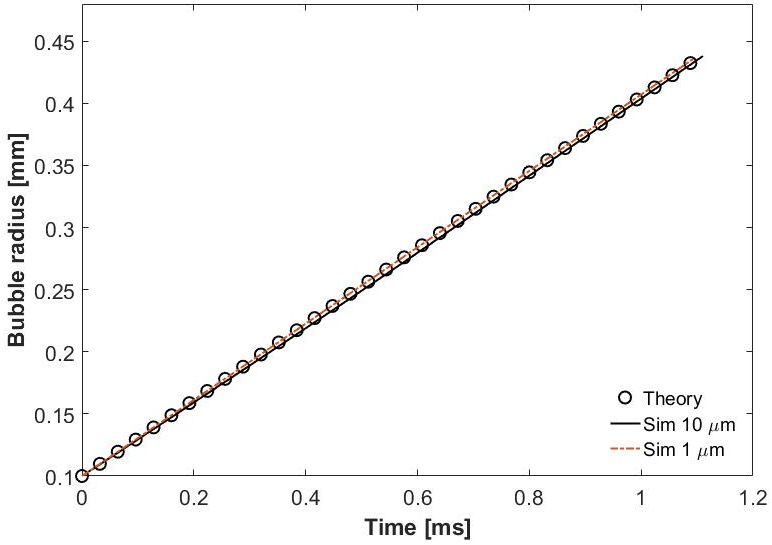

Figure 40: Comparison theory versus simulation of adiabatic bubble growth with a constant evaporative mass flux.

phases have F_l equal to 0 and 1, respectively. The interface-cells (cells with an interface) have liquid volume-fractions in the range of 0 to 1. The initial bubble radius is 0.1 mm and at 0.5 ms the bubble radius is 0.25 mm. The contours of volume-fraction indicate that the bubble maintains its shape during the growth process. In addition, the results in Figure 41(a) show that interface lies within one cell since only one cell has a volume-fraction between 0 and 1 along the interface. A comparison between the contours in Figure 41(a) and 41(b) shows that a decrease in the grid cell size improves the resolution around the interface; a grid cell size of 1 μm creates an almost unperceived division between the liquid and the vapor phases. These results indicate that a peak in the mass source terms at the interface-cells provides a stable bubble growth simulation; this is true for any grid cell size.

Figure 42 shows the velocity vectors on both phases at a simulation time of 0.5 ms for a bubble that grows with a evaporative constant mass flux. The data generated by the numerical solver was exported to a graphics software to provide a better visualization of the flow behavior. The numbers around the contour give coordinates in millimeters. The vapor velocity is small because the bubble remains static during phase change. The liquid

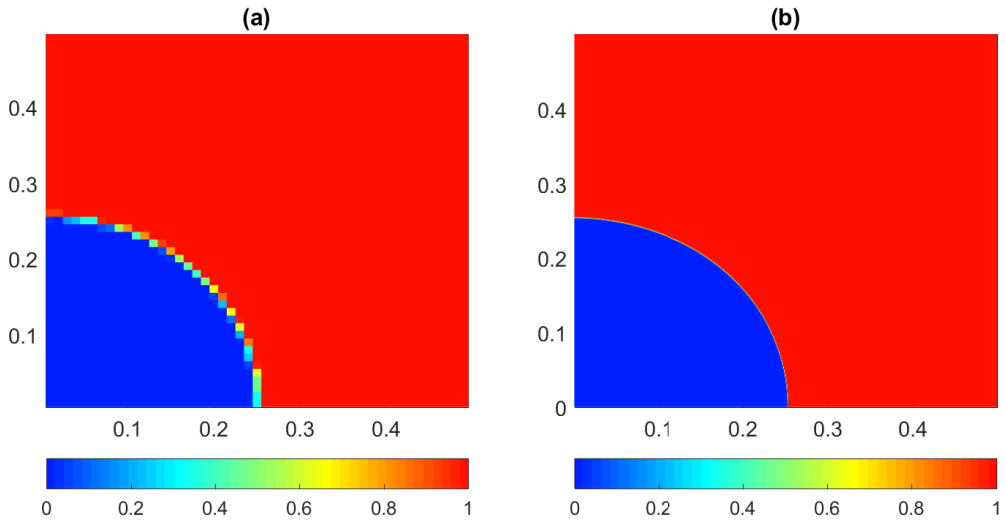

Figure 41: Liquid volume-fractions at 0.5 ms in adiabatic bubble growth. Two different grid cell sizes: (a) $\Delta S = 10$ μm, (b) $\Delta S = 1$ μm.

velocity is highest at the interface and it decreases as the fluid gets far from the interface due to the increase in the fluid area. At the interface, the liquid velocity is high due to the expansion of the vapor during phase change. For a certain volume of liquid consumed ΔV_l, the equivalent volume of vapor generated ΔV_v is almost 1600 times higher ($\rho_l \Delta V_l = \rho_v \Delta V_v$). Therefore, during the phase change process, the vapor expansion increases the momentum of the surrounding liquid. In addition, it is worth to notice the influence of the grid cell size on the fluid velocity around the interface. The vapor velocity near the interface is high with a grid cell size of 10 μm, and it is relatively small with a grid cell size of 1 μm. This behavior is due to assumed average properties on the interface-cells in the solution of the momentum equation. The results show that this approximation generates nonphysical velocities on the neighboring-cells on the vapor side, which become more notorious with larger grid cell sizes. However, as the grid cell size is refined, the effect of the average properties on the interface-cells is less prominent, and the simulation provides a better representation of jump on the liquid velocity at the interface.

Figure 43 compares the magnitude of the numerical and theoretical velocities on both

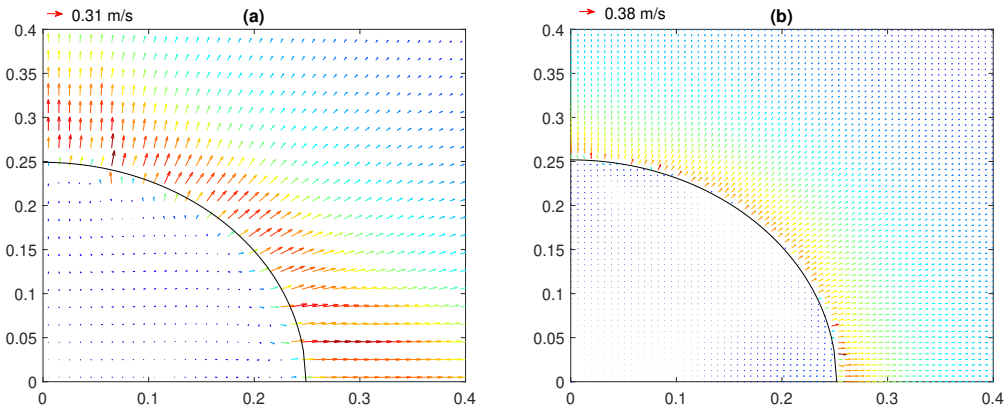

Figure 42: Fluid velocity vectors at 0.5 ms in adiabatic bubble growth with two different grid cell sizes: (a) $\Delta S = 10$ μm, (b) $\Delta S = 1$ μm.

phases. Results in Figure 43(a) show the liquid velocity on the G-cells along the θ direction from $0^\circ \leq \theta \leq 90^\circ$. Results in Figure 43(a) show differences between the numerical and theoretical velocities. The results in Figure 43(b) show the fluid velocity along the radial direction from $0 \leq r \leq 0.5$ mm at $\theta = 0^\circ$. Despite the differences observed in Figure 43(a), the results in Figure 43(b) show excellent agreement for the liquid velocity because the plot considers fluid velocities at $\theta = 0^\circ$. The results in Figure 43(a) show appreciable variations between the numerical and theoretical velocities along the θ direction. However, the numerical and theoretical velocities have the same order of magnitude. The maximum relative error is 16%, and the average relative error is 0.75%. The deviations of liquid velocities oscillate around the theoretical values, which contributes to reduce the magnitude of the average relative error. Maximum deviations occur around 15° and 75°; around these angles, we observed the presences of a large number of interface-cells with small interfaces. Small interfaces are far from the interface-cell center. Therefore, the source of the deviation of the liquid velocities could be related to the estimation of the normal vector at the interface-cell center rather than at the interface. The errors in the estimation of the normal vector affect the surface tension effects at the interface, which creates deviations in the fluid velocities around the interface. The results in Figure 42(b) show that the simulation accurately predicts the variation of the velocity along the radial direction. However, the simulation predicts

a continuous change of the vapor velocity near the interface. The continuous change occurs because the interface-cells have average properties. To capture the jump in velocity at the interface, the simulation should consider ghost velocities at the interface-cells as done by Tanguy et al. [77]. However, it is important to mention that the continuous jump has a minimal effect in the bubble growth rate since the mass transfer model is independent of the velocities near the interface. Therefore, the use of average properties at the interface-cells in the solution of the momentum equation might still provide a reasonable approximation to the bubble growth process.

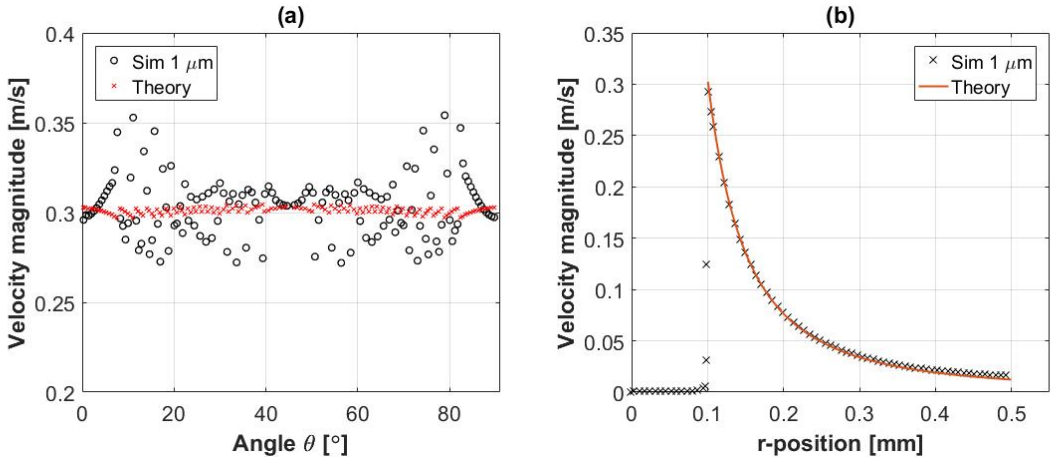

Figure 43: Comparison of the numerical and theoretical velocities after one time step. (a) Near the interface on the G-cells, and (b) on the cells along the radial direction. Grid cell size $\Delta S = 1$ μm, results reported after first time step.

5.4.2 Non-adiabatic Bubble Growth

In non-adiabatic bubble growth, parasitic velocities near the interface might affect the temperature distribution near the interface. Another challenge is the mass transfer computation, which depends on the temperature gradient at the interface. In addition, the accurate prediction of the temperature distribution near the interface requires a correct computation of the interface-cells temperature.

The simulation of a non-adiabatic bubble was performed for a time of 0.1 ms. The initial bubble radius was $R_0 = 0.1$ mm. The wall superheat was 5 K and the properties correspond

to water at 1 atm. Similar times of simulation and initial conditions were adopted in previous numerical works on bubble growth [3, 6, 10]. Scriven's solution Eq. (114) gave the initial temperature field. The objective is to evaluate if the developed methods correctly simulate the phase change process with a moving interface. These methods include: (i) estimation of the mass transfer with the one-cell method, (ii) estimation of the interface-cell temperatures, (iii) interface-cells segregation algorithm to avoid the deformation of the interface, and (iv) numerical solver customization.

A comparison between the numerical and theoretical bubble radius evaluates the accuracy the interfacial heat and mass transfer. The numerical bubble radius assumes a spherical bubble. Scriven's solution Eq. (113) gives the theoretical bubble radius. Figure 44 compares the computed and the theoretical bubble radius. The computational domain had grid cell sizes of 1, 0.6 and 0.2 μm. The simulations ran in parallel processing and the grid was refined near the interface to reduce the cost of the simulation. The results in Figure 44 indicate that the accuracy of the simulation improves with smaller grid cell sizes. The relative error in bubble radius at the end of the simulation is 6.2, 3.0, and 0.2% with grid cell sizes of 1, 0.6, and 0.2 μm, respectively. The improvement is due to the higher grid resolution in the thermal film obtained with smaller grid cells. The number of cells in the thermal film at the beginning of the simulation is 12, 20, and 80 for grid cell sizes of 1, 0.6, and 0.2 μm, respectively. Interestingly, similar to the simulation of planar interface evaporation, the simulation of a spherical bubble required at least 80 cells to capture the thermal film near the interface. Another important observation is that large grid cell sizes lead to smaller bubble growth rates. These results indicates that large grid cells size lead to a faster growth of the thermal film relative to the theory. The faster growth creates smaller mass fluxes because the magnitude of the interfacial temperature gradients decreases.

The variation of the temperature gradients along the interface (see Figure 28) could create a non-uniform mass transfer. Moreover, the interface was not smeared, which account for the abrupt change in the thermal properties at the interface. These factors might introduce

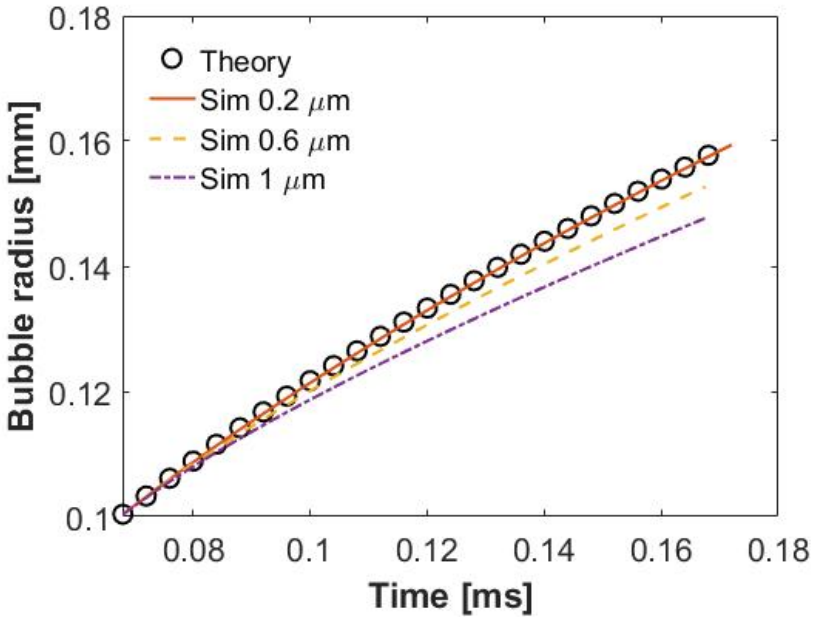

Figure 44: Prediction of the bubble growth rate. Non-adiabatic bubble growth with temperature gradients at the interface.

interface deformations or numerical instabilities as pointed out by previous works. To observe possible deformations of the interface, the bubble shape was analyzed at different times. Figure 45 shows the liquid volume-fractions at times of 0.11 and 0.16 ms for a grid cell size of 0.6 μm. The results distinguish interfaces without appreciable deformations. Moreover, the bubble conserved its shape throughout the simulation. In addition, the numerical solution was stable since the residuals for the governing equations reached the convergence criteria. These results indicate that the variations in the estimation of the temperature gradient generated by the one-cell method are insignificant, and therefore they have no influence on the shape of the interface. In addition, the results indicate that fixing the temperature of the interface-cells (to account for the sharp interface) and the declaration of mass transfer only at the interface-cells lead to a stable simulation.

Figure 46 shows the velocity distribution on both phases at two different instances. Figure 46(a) shows the velocity distribution at 0.11 ms, and Figure 46(b) shows the velocity distribution at 0.16 ms. A comparison between the two figures shows that the liquid veloc-

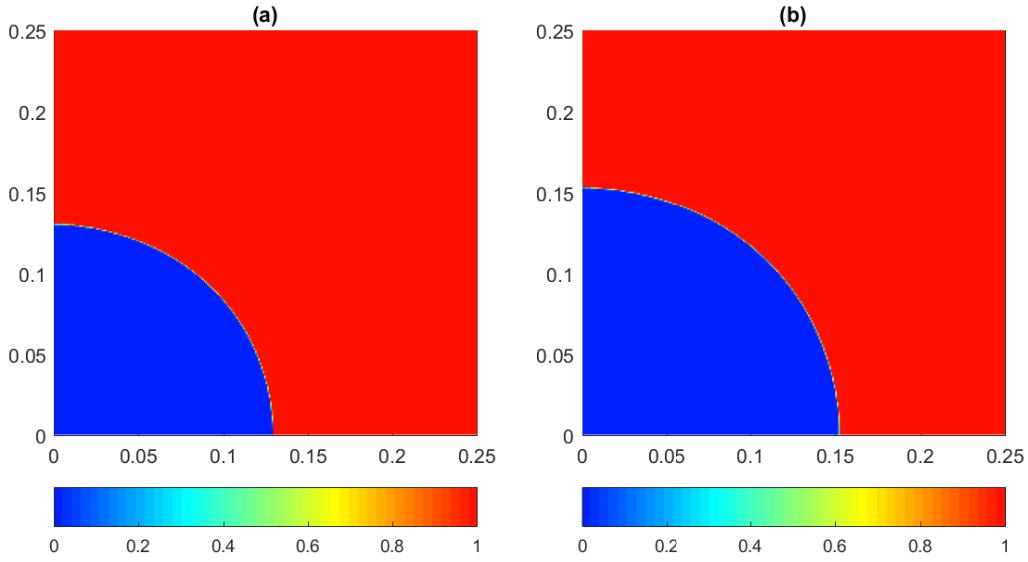

Figure 45: Liquid volume-fractions in non-adiabatic bubble growth at two simulation times. Grid cell size 0.6 μm. (a) t =0.11 ms (b) t =0.16 ms. Axis dimension is millimeters.

ity decreases with time. This occurs because as the bubble grows, the mass flux becomes smaller due to the increase of the thermal film thickness. The results also indicate maximum liquid velocities of 0.8 m/s and 0.68 m/s at 0.11 ms and at 0.16 ms, respectively. These magnitudes are higher than the theoretical values of 0.56 m/s and 0.46 m/s at 0.11 ms and at 0.16 ms, respectively. As shown in Figure 46(a), the liquid velocity distribution along the interface follows an oscillatory trend, and therefore higher numerical velocities relative to theoretical data is an expected trend. Nevertheless, the results indicate that simulation predicts velocities that are in the same order of magnitude as in the theory, and that the oscillations in the velocity magnitude occurs only on certain regions along the interface.

Deviations in the velocity of the liquid might affect the temperature distribution due to the convective transport. However, conduction also governs the heat transfer distribution near the interface. In the present case of analysis, the convective transport plays an important role since Peclet number $Pe = \rho_l v_l / \left(\frac{k_l}{c_l}\frac{1}{\Delta s}\right)$ changes from 2.6 to 1.7. This implies that initially, when the bubble is small, convection governs the heat transfer because the mass

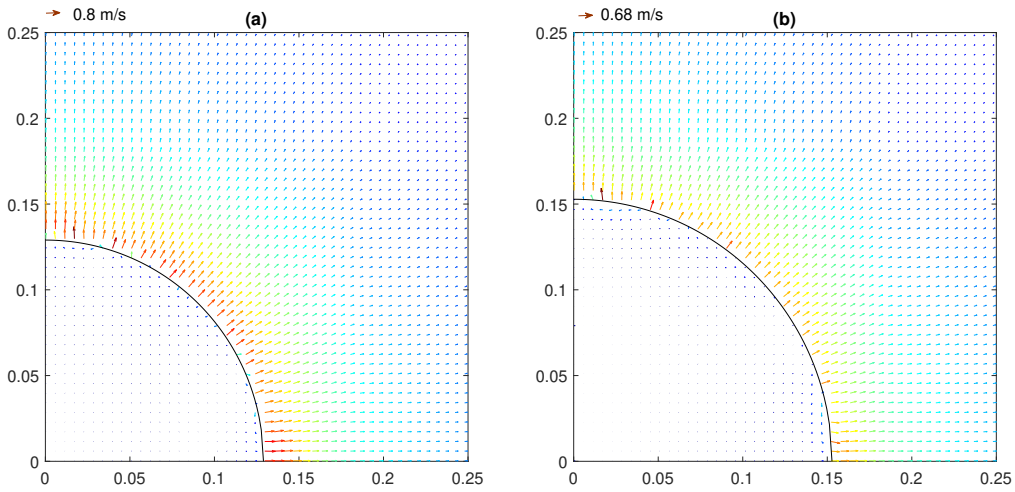

Figure 46: Velocity vectors in non-adiabatic bubble growth. Grid cell size 0.6 μm. (a) t =0.11 ms (b) t =0.16 ms. Axis dimension is millimeters.

flux is high. As the bubble continues to grow, the convective effect becomes less significant and conduction begins to control the heat transfer near the interface. Figure 47 shows the temperature distribution on both phases at two different times. The vapor phase remains at the saturation temperature of 373.15 K, whereas the liquid phase has a temperature that changes from 373.15 K to 378.15 K. In the contours, the interface-cells have the saturation temperature instead of a fixed temperature for clarity purposes. As mentioned earlier, the thermal film forms due to the thermal interaction between the interface and the liquid. The results show that the temperature distribution is uniform along the interface, which proves a correct fixing of the interface-cells to account for a sharp interface. Another important fact is that the thermal film remains thin throughout the bubble growth process. The thickness of the thermal film changes from 12 μm to 20 μm in the simulation. The thermal film is not consumed during bubble growth, but it moves along with the interface due to a convective transport. At the same time, the thickness of the thermal film increases with time due to the diffusive transport.

Results in Figure 47(b) indicate that as the bubble grows, slight deformations of the thermal film appear near the boundaries. The reason of these variances is an increase in

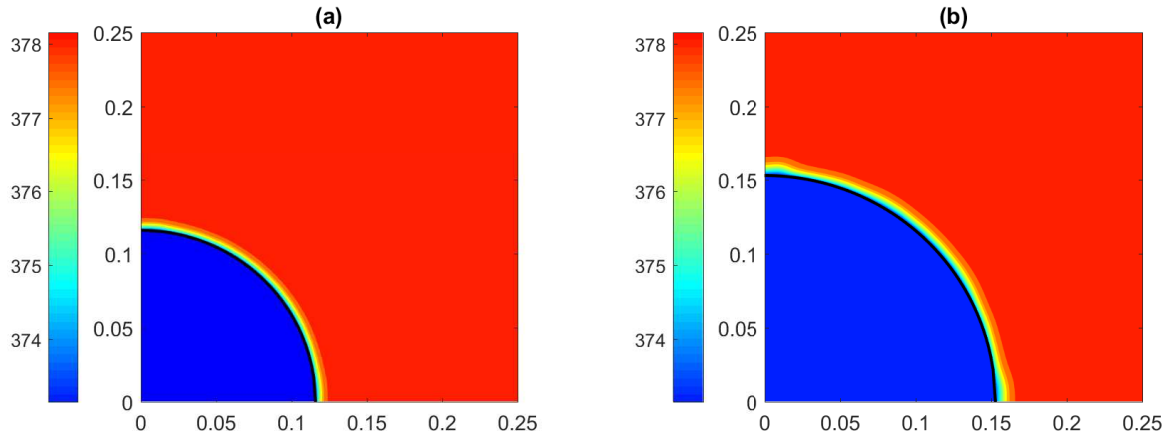

Figure 47: Temperature distribution in non-adiabatic bubble growth at two simulation times. Grid cell size 0.6 μm. (a) t =0.11 ms (b) t =0.16 ms. Axis dimension is millimeters.

the liquid velocity on these regions. Figure 43(a), which plots the velocity distribution near the interface along θ direction, shows clear evidence of the increase in the liquid velocity on these regions. As observed in Figure 43(a), the changes in the liquid velocity occur in adiabatic bubble growth; therefore, the errors are unrelated to the methods proposed to estimate the mass transfer or the temperature of the interface-cell. Such a trend could be due to secondary effects of errors in normal vectors around 15° and 75°. Different to other regions, the velocity near the boundaries does not oscillates around the theoretical velocity. Therefore, these variations in velocity enhance the convective heat transfer, which causes the deformation of the thermal film. However, results indicate that this deformation of the thermal film has a minimal influence in the mass transfer and bubble growth rate.

5.5 Simulation of Nucleate Boiling

This section discusses results of the simulation related the thermal and dynamic fluid behavior during nucleate boiling. The simulation of nucleate boiling requires accurate estimations of the interfacial temperature gradients and a precise estimation of the heat transfer and evaporation rates due to microlayer effects. In addition, the simulation requires proper definition of mass transfer near the contact line to avoid interface breaking or deformations.

The simulation in the present work includes various features: (i) direct estimation of the mass transfer based on the interfacial temperature gradients with the proposed external-software method (see section 2.1.3), (ii) interface-cells with an interpolated temperature with the proposed linear temperature profile method (see section 2.2.2), (iii) a sharp interface that lies within one cell with mass transfer defined only at the interface-cells with the proposed interface-segregation method (see section 2.3), (iv) computation of surface tension with smoothed volume-fractions, and (v) microlayer evaporation at the contact line. In the simulation of nucleate boiling, the time step was variable and determined with a maximum Courant number of 0.25.

5.5.1 Microlayer Heat Transfer and Evaporation

Figure 48 shows the microlayer trends obtained with the microlayer model (described in section 3.2). Figure 48(a) shows the thickness and heat flux along the microlayer length, and Figure 48(b) shows the variation of the integrated heat flux and interface slope along the microlayer length. The model predicts a flat interface (adsorbed layer) near ξ =0. The heat flux experiences an abrupt change near at the transition between the adsorbed layer and the microlayer; the heat flux is 0 at ξ =0 and has a maximum value of 8.7 kW/cm^2 at ξ =0.026 μm. In this interval, the thickness of the flat interface changes from 0.6 nm at ξ =0 to 2.77 nm at ξ =0.026 μm, and the slope changes from 0° at ξ =0 to 14.8° at ξ =0.026 μm. After ξ =0.026 μm, the heat flux decreases to value of 1.8 kW/cm^2 at ξ =0.5 μm. At ξ =0.5 μm, the microlayer slope become asymptotic and equal to 29°. The relatively low heat flux and the asymptotic microlayer slope at ξ =0.5 μm indicates the end of the microlayer effects. At ξ >0.5 μm, the heat transfer and the evaporation depends mainly on the distance between the interface (with a temperature T_{sat}) and the heated surface. The integrated heat flux continuously increases with the microlayer length with a value of 17.1 W/m at 0.5 μm.

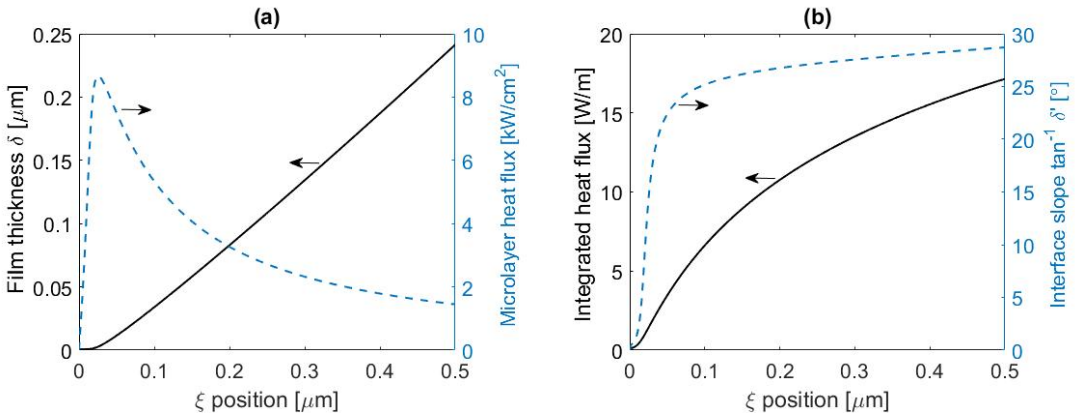

Figure 48: Microlayer characteristics obtained with the microlayer model, (a) microlayer shape and heat flux, (b) microlayer integrated heat flux and slope. $\Delta T_w = 6.2$ K, $A = 8.5 \times 10^{-21}$ J.

5.5.2 Mesh Sensitivity Analysis and Validation

To show the ability of the proposed approach to reproduce experimental data on nucleate boiling, the present work compares the numerical data against the experimental data of Siegel and Kenshock [81] on bubble growth rate. A few numerical works have used the experimental data of Siegel and Kenshock [81] to validate their simulation [3,17]. The operating conditions consider a wall superheat of 6.2 K with water at 1 atm. In addition, the simulation considers grid cell sizes of 20, 16, and 12 μm. Moreover, the simulation evaluates the effect of first and second order approximations on the estimation of interfacial temperature gradients.

Figure 49 compares the numerical results against the experimental data of Siegel and Kenshock [81] for bubble growth during one cycle. Figure 49(a) shows the effect of the grid cell size, and Figure 49(b) shows the effect of a second order approximation. Experiments indicate a departure bubble radius of 1.05 mm whereas the simulation predicts a departure radius of 1.12 mm. The small difference might be due to assumption of bubble departure when the interface reaches $r = 0$ m instead of considering the effect of the cavity. At the time of departure in the experiments (0.032 s), the experimental bubble radius is 1.05 mm and the numerical bubble radius is 1.02 mm. The grid cell size influences the bubble growth rate at the initial stage of bubble growth (when the time is less than 0.021 s), but its influence

on the departure bubble diameter is minimal. At the initial stage, superheated liquid covers the interface almost entirely and the mass transfer governs the bubble shape. Near bubble departure, superheated liquid covers just a small section of the interface and surface tension governs the bubble shape. Therefore, the grid cell size has a significant effect on the bubble shape at the initial stage of the bubble since the mass transfer depends on the interfacial temperature gradients. Results in Figure 49(b) show the effect of the order of approximation on the estimation of the interfacial temperature gradients. To our knowledge, this is the first report that evaluates the effect of the order of approximation on the simulation of nucleate boiling. The figure shows results of a first order approximation with a grid cell size of 12 μm and a second order approximation with a grid cell size of 16 μm. Results indicate that the order of approximation is an important parameter on the simulation of nucleate boiling. Relative to the second order approximation, the first order approximation generates slower bubble growth rates despite the smaller grid cell size. Relative errors (relative to experiments) at 0.032 s are 11.7 and 3.4% with first and second order approximations, respectively. The second order approximation generates faster bubble growth rates due to the higher and more accurate temperature gradients (as discussed in section 5.2.3).

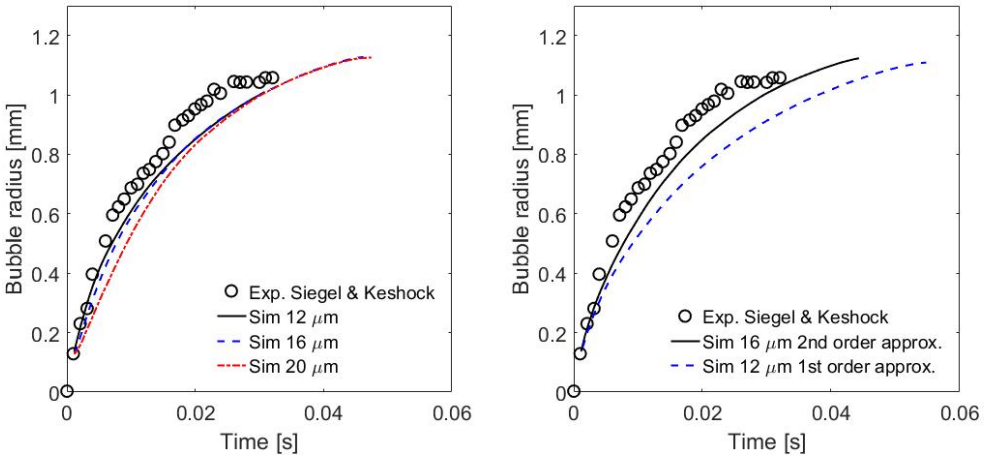

Figure 49: Comparison numerical results versus experiments; bubble growth rates in nucleate boiling. (a) Effect of the grid cell size, (b) effect of the order of approximation on the estimation of interfacial temperature gradients.

5.5.3 Dynamic and Thermal Fluid Behavior During Nucleate Boiling

Figure 50 shows the liquid-volume fractions at different instances during a bubble growth cycle. Previous simulations of nucleate boiling define a transition region around the interface or use an interface sharpening equation to avoid interface deformations. However, in the present simulation the interface lies within one computational cell and the proposed interface-cell segregation algorithm conserves the interface sharpness. Another important fact is that the simulation computes surface tension effects with smoothed volume-fractions to ensure proper bubble shapes. To our knowledge, this is the first report that simulates nucleate boiling with VOF by keeping the interface within one computational cell. Results show that the interface remains sharp and with a semi-circular shape throughout the ebullition cycle. The initial conditions consider a bubble of radius R_0 =0.12 mm. The mass transfer at the interface makes the bubble to grow by changing the volume-fractions at the interface-cells. During the initial stage of bubble growth, the liquid recedes (receding interface) and the interface velocity depends on the magnitude of the mass transfer. During the necking stage (t >0.026 s), the liquid advances (advancing interface) and the interface velocity depends on the magnitude of the fluid velocities. At t =0.043 s the bubble departs, which occurs when the contact line reaches the radial location r = 0 m. After bubble departure, the bubble moves in the vertical direction merely by the effect of the fluid velocity (no mass transfer occurs at the interface). In addition, results indicate that the contact line moves during the bubble growth process; the radial location of the contact line is 0.36, 0.47, and 0.15 mm at times of 8, 26, and 43 ms, respectively. The contact line moves despite the non-slip boundary condition over the surface; the reason is that the interface-tracking algorithm moves the contact line based on the mass transfer and the vapor velocity at the contact-line cell center.

Figure 51 shows the velocity vectors on the liquid at different times during one bubble cycle. The color and size of the vectors indicates the magnitude of the fluid velocity. A sharp interface delineates the interface shape in the simulation. The simulation uses volume-

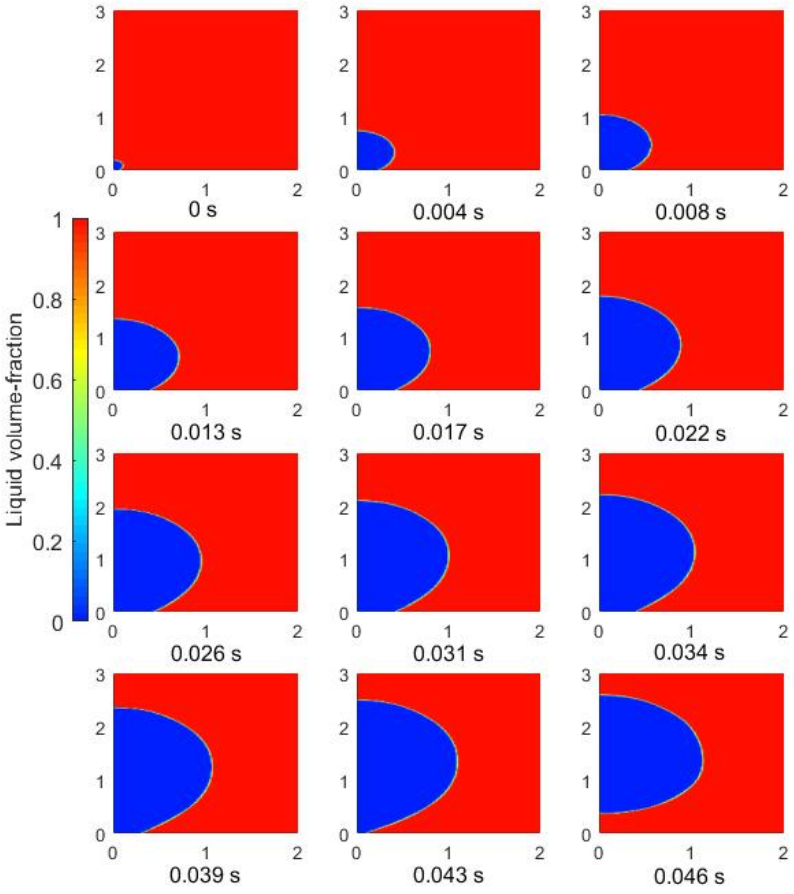

Figure 50: Liquid volume fractions with a sharp interface during bubble cycle (axis dimensions in mm).

fractions to reconstruct the sharp interface. It is important to highlight that in the simulation the mass transfer model links the fluid velocities with the temperature gradients at the bubble-edge. The interfacial temperature gradients dictate the mass transfer, and the mass transfer dictates the liquid velocity at the interface. Results show that the liquid travels in a positive direction (radial and axial) when the interface recedes. In addition, results indicate an increase in momentum of the fluid near the interface and a reduction of the liquid velocity near the surface. The direction of the flow and the increase in momentum occur because the vapor expansion moves the liquid away from the interface. The reduction of the liquid velocity near the heated surface is due to the non-slip boundary condition, which reduces

the fluid velocity to zero at the surface. At t =0.013 s, the liquid near the surface moves in the positive radial direction and a fluid circulations appears at the middle section of the bubble-edge. Results show that the fluid circulation becomes stronger with time and that the fluid circulation changes the direction of the flow near the contact line region. At t =0.026 s, the liquid moving in the negative radial direction makes the interface to advance. As the interface advances, liquid velocity near the contact line increases and the liquid travels towards the surface. After bubble departure, accelerated liquid occupies the gap between the bubble-edge and the heated surface. The departed bubble creates a large fluid circulation on the lateral side with strong currents underneath the bubble.

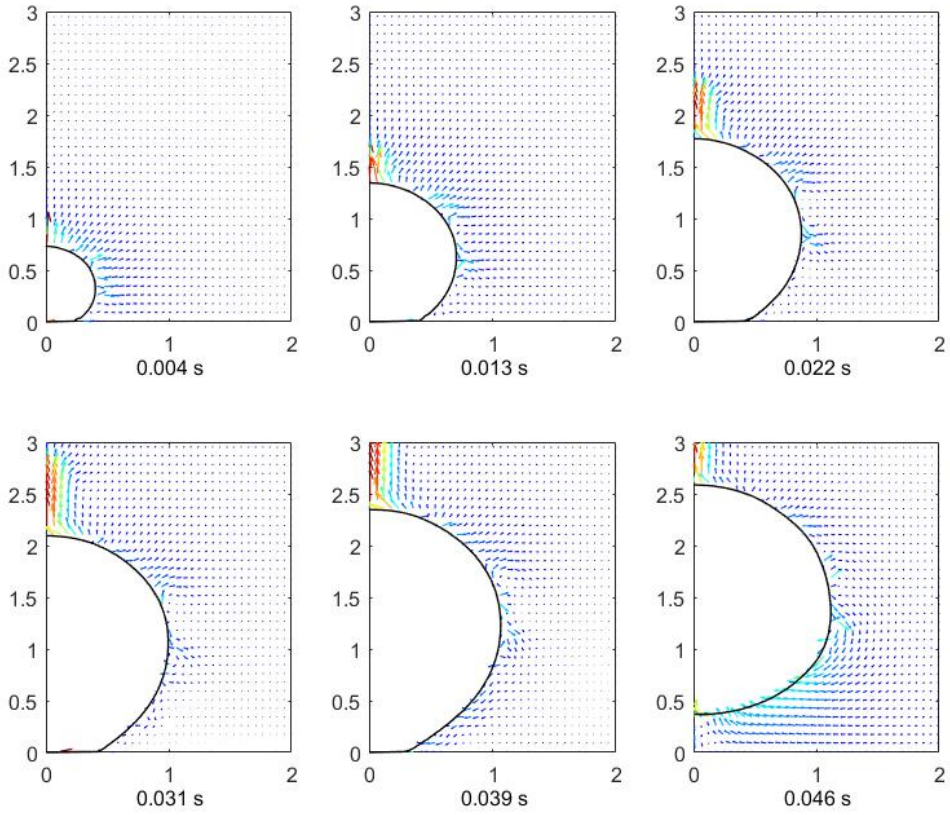

Figure 51: Liquid velocity vectors during one cycle bubble growth (axis dimensions in mm). Interface shape reconstructed from volume fractions.

Figure 52 shows the shear stress over the surface at different times during a bubble cycle.

A shear stress different to zero implies a region affected by the interfacial phase-change process and interface curvature (influence region). Results reveal that the vapor expansion during the initial stage of the bubble generates a large influence region. The length of the influence region decreases as the interface recedes. Such a trend occurs because the liquid recirculation (formed due to curvature effects) counteracts the effect of the vapor expansion. The shear stress keeps decreasing after the interface begins to advance (t >0.026). However, in the range of time 0.034 to 0.046 s the interface moves much faster, which significantly increases the length of the influence region. The maximum length of the influence region occurs as the interface recedes (rewetting process). Results indicate a maximum influence region of 2.2 mm from the central axis, which corresponds to a total length of 4.4 mm. Therefore, results reveal that the influence region is about $2D_b$, where D_b is the departure bubble diameter (the numerical departure bubble diameter is 2.24 mm, see Figure 9). Such a length of the influence region is in agreement with experimental data of Hsu and Graham [82].

Figure 53 shows the temperature distribution during the complete ebullition cycle. In the simulation, the proposed sharp interface model adjusts the temperature of the interface-cells to account for the interface position, the interface temperature, and to eliminate the use of average properties on the interface-cells. Results reveal that the interface temperature affects the liquid temperature. In the range from 0 to 1.7 ms, a thermal film forms on the liquid that changes the temperature from T_w at the surface to T_{sat} at the interface. Such a behavior creates a meniscus region with large temperature gradients due to the close proximity between the interface and the heated surface. In the meniscus region, the interface cools down the liquid around the bubble-edge. The heat moves with the interface due to convection heat transfer. Therefore, the correct temperature distribution on the meniscus region requires precise liquid velocities near the bubble-edge (such precise liquid velocities come from the accurate estimation of the interfacial mass transfer with the proposed model). At $t > 2.6$ ms, liquid begins to rewet the surface. Results reveal that the rewetting process leaves a layer of liquid with a low temperature over the surface. Results reveal a reduction of

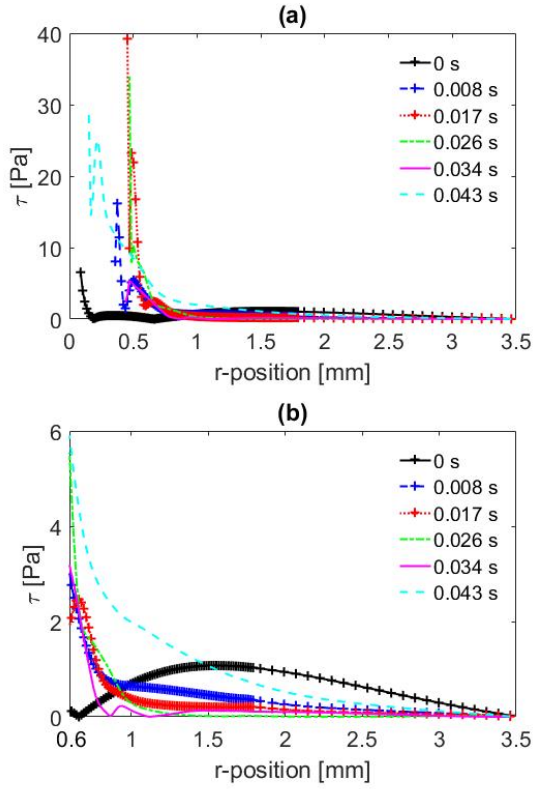

Figure 52: Influence region given by surface shear stress. (a) Shear over complete surface length, (b) shear over surface in the range of 0.6 to 3.5 mm. Lines with a + sign correspond to the receding stage.

the thickness of the thermal layer near the interface during the advancing process, which is due to cold liquid traveling towards the surface by means of the interface curvature (transient conduction mechanism). The bubble departs at 46 ms and colder liquid occupies the space between the interface and the heated surface. Hereafter, the bubble moves upwards and convection heat transfer moves the heat up, which increases the thickness of the thermal layer. After 0.01 s (the estimated waiting time), the next bubble appears and a new cycle starts.

Figure 54 shows the variation of the local heat transfer coefficient along the surface where $h = q''_{z=0}/\Delta T_w$ gives the local heat transfer coefficient. The results consider the radial position of the contact line (ξ_{cl}) as a reference point; in addition, the results in the figure omit the heat transfer coefficient related to microlayer and transition region at the contact (the

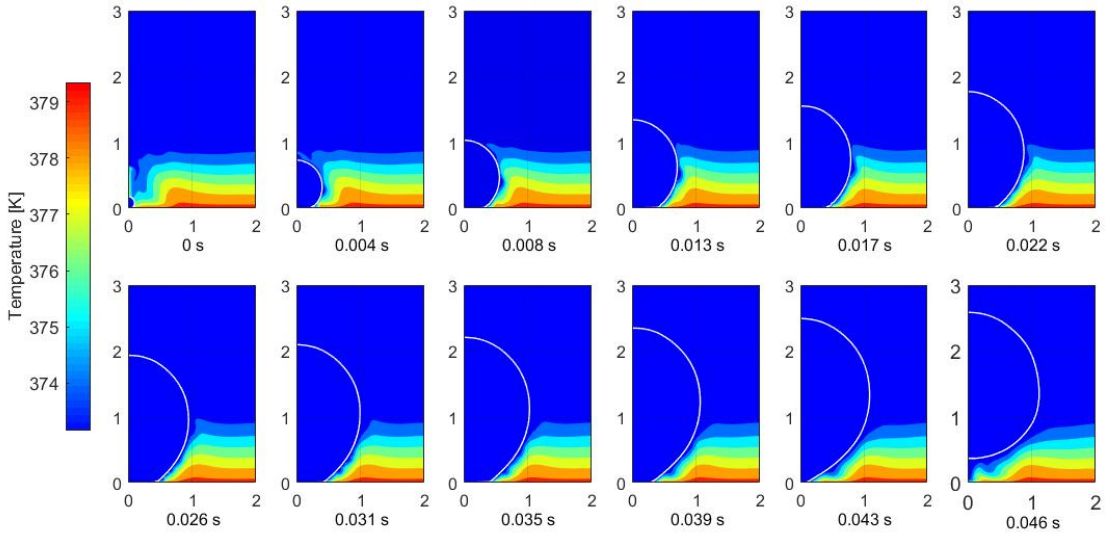

Figure 53: Temperature variation during a bubble cycle (axis dimensions in mm). Interface shape reconstructed from volume fractions.

simulation considered microlayer effects, but the graph omits the heat transfer coefficient at the contact line for clarity purposes). It is important to highlight that the sharp interface model used in the simulation allows access to the heat transfer coefficient near the contact line and that the correct interface treatment gives accurate temperatures on the neighboring-cells. Results show that before the patching of the bubble at $t = 0$ s, the heat transfer is low but it is non-uniform along the surface due to the effect of the previous bubble. At $t = 0$ s the bubble appears and the interface begins to recede due to the mass transfer at the interface. During the receding stage, a large heat transfer coefficient appears near the contact line. Heat transfer coefficients of the order of 1x10^5 W/m^2-K appear near the contact line and the magnitude is still significant at a distance of 200 μm from the contact line. The increase in the heat transfer coefficient is due to the development of the meniscus region and it is unrelated to microlayer effects (microlayer effects appear underneath the bubble and the results show only heat transfer on the macroregion). During the advancing interface ($t >$0.026 s), the heat transfer coefficient increases due to the layer of liquid left by the contact line and to cold liquid traveling towards the interface. In addition, during the

rewetting process, the heat transfer coefficient far from the interface decreases since liquid with a higher temperature gets in contact with the surface.

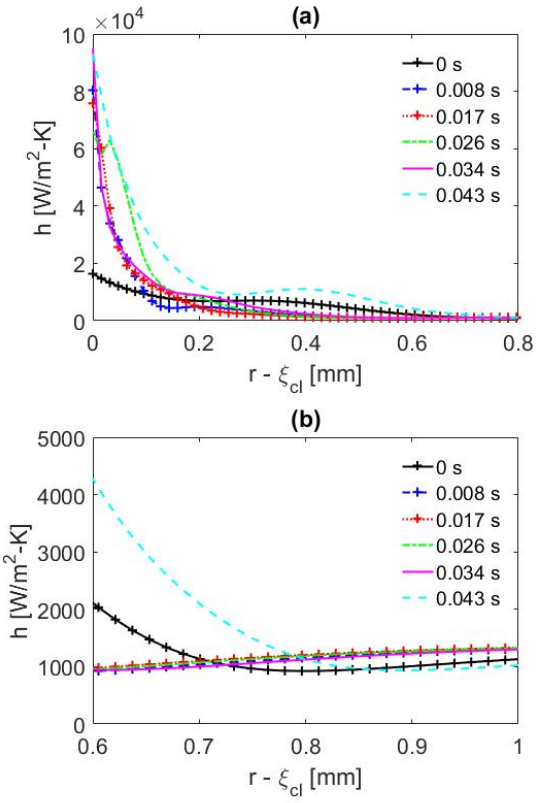

Figure 54: Influence region given by local heat transfer coefficient over the surface. (a) radial location in the range between 0 to 0.8 mm, (b) radial location in the range of 0.6 to 0.7 mm. Lines with a + sign correspond to the receding stage.

6 Conclusions and Future Work

The work accomplished within the present thesis consists of two mains sections: (i) theoretical analyses of fundamental problems and development of numerical methods to reproduce theoretical data, and (ii) simulation of nucleate boiling with a sharp interface and with mass transfer that depends on temperature gradients at the interface.

6.1 Theoretical Analyses and Development of Numerical Methods

The work considered theoretical analyses of fundamental problems of phase-change such as planar interface evaporation and spherical bubble growth. The theoretical analyses provided fundamental knowledge on the mechanisms of phase-change including the dependence of mass transfer with interfacial temperature gradients, the fluid behavior during-phase change, and the role of the fluid properties (latent heat of evaporation, fluid density, thermal conductivity, etc.) In addition, the theoretical analyses provided a reference for the development of the simulations. We proposed eight Stefan-type cases of planar interface evaporation that consider heat transfer on both liquid and vapor phases and that accounted for the effect of density difference between vapor and liquid phases.

The simulation of planar interface evaporation indicated the need to develop models to consider the interface temperature and to compute the mass transfer with interfacial temperature gradients along the interface normal direction in order to generate accurate temperature fields and interface displacements. We developed numerical methods to include the interface as an internal boundary condition with the saturation temperature and to compute the evaporative mass flux based on the temperature gradients at the interface. Results indicated that the simulation reproduced the proposed Stefan problems with heat transfer on both phases. A comparison between the proposed theoretical models and the numerical results showed excellent agreement in terms of temperature distribution at specific times and interface displacement.

The simulation of spherical bubble growth in superheated liquid showed the need to develop methods to compute the evaporative mass flux with interfacial temperature gradients that consider the interface orientation and location. In addition, the simulation requires special numerical methods to account for inclined (non-horizontal) interfaces that have the saturation temperature. Moreover, special methods should declare mass transfer on interface-cells (computational cells with an interface) in a way that preserves the interface sharpness (avoid interface breaking and deformations). The proposed one-cell method computes mass transfer based on the interface normal vector and uses the temperature of a single cell to compute the temperature gradients at the interface. In addition, we proposed a method to interpolate and fix the temperature of the interface cells based on the temperature gradients at the interface. The proposed method to interpolate the temperature of the interface-cells assumes a linear temperature profile approximation along a direction normal to the interface. Finally, we proposed a segregation interface-cells method to preserve the interface sharpness. The segregation method declares mass transfer just at the interface-cells that lie next to vapor cells. The proposed methods allowed the simulation of spherical bubble growth with a sharp interface (interface within one cell) with accurate temperature gradients at the interface. To our knowledge, this is the first work that simulates bubble growth with a sharp interface and mass transfer only at the interface cells withing the volume-of-fluid interface tracking method. Results of the simulation indicated excellent agreement with theoretical data on spherical bubble growth (0.2% relative error on interface displacement with at grid cell size of 0.2 μm).

6.2 Simulation of Nucleate Boiling

The simulation of planar interface evaporation and spherical bubble growth provided the framework to simulate nucleate boiling. In addition, the simulation requires an external model to account for the contribution of microlayer evaporation considering the molecular forces between the fluid and the surface at the contact line. Moreover, the non-uniform

temperature around the interface requires a precise estimation of the interfacial temperature gradients to capture the variation of mass transfer along the interface. An external model that considered the solution of a fourth order differential equation determined the contribution of the contact line evaporation. In addition, we proposed a method to perform the computation of the temperature gradients in MATLAB. This mathematical software has advanced interpolation functions that allowed the precise estimation of temperature gradients with first and second order approximations. Results showed that mass transfer with second order approximations leads to more accurate bubble growth rates (relative to experimental data). Results indicated that the interpolation of the interface-cells temperature is essential to generate proper temperatures at neighboring cells near the contact line. Additionally, results showed that the simulation of a sharp interface (interface within one cell) requires a segregation algorithm to prevent the interface breaking or deformation.

The simulation of nucleate boiling with mass transfer based on temperature gradients at the interface, with an interface at the saturation temperature, and with a sharp interface (interface within one cell) generated bubble growth rates that compare well with experimental data on bubble growth rate. Results showed a maximum error of 3.4% relative to experimental data on bubble growth rate over a heated surface. In addition, the numerical simulation with a sharp interface and with an interface at the saturation temperature gives access to the heat transfer near the contact line. This information revealed high heat transfer coefficients of the order of 1x10^5 W/m^2-K up to 200 μm from the contact line. Numerical data on the wall shear stress over the surface indicated that the nucleation of a bubble has an influence region of two times the departure bubble diameter.

In summary, results indicated that the simulation described in the present work is a valuable tool to analyze the heat transfer mechanisms in bubble nucleation. The proposed methods improved simulations of multiphase flows by describing methods to compute mass transfer with temperature gradients at the interface, to include the interface saturation temperature, and to preserve the interface sharpness in the modeling of sharp interfaces

(interface within one cell).

6.3 Potential Future Projects

Future work should use the developed model to analyze the heat transfer mechanisms of nucleate boiling at different conditions. In addition, work should be done to extend the capability of the proposed numerical model to cover different aspects of heat transfer in boiling.

At the present, the simulation showed good approximation to experimental data of bubble growth in water at 1 atm and at wall superheat level of 6.2 K. Future work should simulate nucleate boiling with various fluids and operating conditions. The main barriers lie in the estimation of the mass transfer due to microlayer effects at such conditions. The agreement of the numerical results against experimental data on bubble growth rates will increase the confidence on the developed simulation to predict heat transfer performance of a nucleating bubble. In addition, such simulations could be used to quantify the contribution of the different heat transfer mechanisms including microlayer heat transfer.

Efforts should be put to include the solid heated plate in the simulation. At the present, the simulation assumes a wall with a constant temperature as the boundary condition to account for the heated plate. However, the heat transfer on the solid plate due to the contact line and the meniscus region is an area that requires special attention. The step of adding the solid plate with a constant heat flux is straightforward since simulations in ANSYS-Fluent impose a continuous heat flux approximation to model the contact between the fluid and the solid plate. However, simulations with a solid plate should consider the effect of the change in the surface temperature near the contact line. The microlayer model uses the temperature of the surface as an input to determine the amount of liquid that evaporates at the contact line. Therefore, the simulation with a solid plate should continuously communicate the overall simulation with the microlayer model to determine the contact line evaporation based on the surface temperature at the contact line.

A second improvement is to perform 3D simulations. At the present, the developed numerical model assumes a 2D axisymmetric condition. The 2D axisymmetric condition considers a cylindrical domain with zero temperature gradients along the angular direction. Such assumption is valid in cases where the flow shows symmetrical behavior referent to the z-axis at center of the heated surface. 3D simulations will allow the analysis of more complex geometries such as boiling in porous media, boiling in open microchannels with tapered manifolds, boiling in microstructures, etc. The extension of the proposed models to 3D simulations is straightforward since the proposed methods are compatible with conventional discretization schemes. However, 3D simulations requires good computer resources due to the large number of computational cells.

The numerical model could be extended to simulate phase-change of multicomponent fluids. Simulations of boiling or droplet formation in multicomponent fluids are important in the fields of industrial refrigeration and microreactors. The simulation of multicomponent fluids will require equations for the species transport. The software ANSYS-Fluent provides a module to solve such equations. The effect of the species on the mass transfer requires additional considerations. However, the mass transfer might depend the concentration gradients and temperature gradients at the interface, which are easy to compute with the proposed one-cell method or the method that uses MATLAB to interpolate values at reference points.

References

[1] Dhir, V. K., Warrier, G. R., and Aktinol, E., 2013. Numerical Simulation of Pool Boiling: A Review. *J. Heat Transf.*, 135(6), pp. 61502-6150217.

[2] Udaykumar HS., and Shyy W., 1995. Simulation of Interfacial Instabilities During Solidification I. Conduction and Capillarity Effects, *International Journal of Heat and Mass Transfer*, 38, 2057-2073.

[3] Sato Y., and Ničeno, B., 2013. A sharp-interface phase change model for a mass-conservative interface tracking method, *Journal of Computational Physics* 249, 127-161.

[4] M. Alpaz., A. Bilgesu., O. Tutkun., 1988. The origin of the equation for the interfacial coefficient of heat transfer, *Commun. Fac. Sci. Univ. Ank. Serie B*, 34 113-128.

[5] Esmaeeli A., and Tryggvason G., 2004. Computations of film boiling. Part I: numerical method, *International Journal of Heat and Mass Transfer*, 47(25), 5451-5461.

[6] Shin S., and Choi B., 2016. Numerical simulation of a rising bubble with phase change, *Applied Thermal Engineering*, 100, 256-266.

[7] Ling K., Li, Z.Y., and Tao, W.Q., 2014. A direct numerical simulation for nucleate boiling by the VOSET method, *Numerical Heat Transfer, Part A: Applications* 65, 949-971.

[8] Juric, D. and Tryggvason, G., 1998. Computations of boiling flows, *International Journal of Multiphase Flow*, 24(3), pp.387-410.

[9] Hardt, S. and Wondra, F., 2008. Evaporation model for interfacial flows based on a continuum-field representation of the source terms. *Journal of Computational Physics*, 227(11), pp.5871-5895.

[10] Kunkelmann, C. and Stephan, P., 2009. CFD simulation of boiling flows using the volume-of-fluid method within OpenFOAM. *Numerical Heat Transfer, Part A: Applications*, 56(8), pp.631-646.

[11] Magnini, M., Pulvirenti, B. and Thome, J.R., 2013. Numerical investigation of hydrodynamics and heat transfer of elongated bubbles during flow boiling in a microchannel. *International Journal of Heat and Mass Transfer*, 59, pp.451-471.

[12] Yang, Y., Pan, L.M. and Xu, J.J., 2014. Effects of microgravity on Marangoni convection and growth characteristic of a single bubble. *Acta Astronautica*, 100, pp.129-139.

[13] Jia, H.W., Zhang, P., Fu, X. and Jiang, S.C., 2015. A numerical investigation of nucleate boiling at a constant surface temperature. *Applied Thermal Engineering*, 88, pp.248-257.

[14] Yazdani, M., Radcliff, T., Soteriou, M. and Alahyari, A.A., 2016. A high-fidelity approach towards simulation of pool boiling. *Physics of Fluids*, 28(1), p.012111.

[15] Liu, Q. and Palm, B., 2016. Numerical study of bubbles rising and merging during convective boiling in micro-channels. *Applied Thermal Engineering*, 99, pp.1141-1151.

[16] Son, G. and Dhir, V.K., 1998. Numerical simulation of film boiling near critical pressures with a level set method. *Journal of Heat Transfer*, 120(1), pp.183-192.

[17] Son, G., Dhir, V.K. and Ramanujapu, N., 1999. Dynamics and heat transfer associated with a single bubble during nucleate boiling on a horizontal surface. *Journal of Heat Transfer*, 121(3), pp.623-631.

[18] Banerjee, D., and Dhir, V. K., 2001, Study of Subcooled Film Boiling on a Horizontal Disc: Part I - Analysis, *J. Heat Transf.-Trans.*, 123(2), pp. 271284.

[19] Son, G., 2001. A numerical method for bubble motion with phase change. *Numerical Heat Transfer: Part B: Fundamentals*, 39(5), pp.509-523.

[20] Li, D. and Dhir, V.K., 2007. Numerical study of a single bubble sliding on a downward facing heated surface. *Journal of Heat Transfer*, 129(7), pp.877-883.

[21] Aktinol, E., and Dhir, V. K., 2012. Numerical Simulation of Nucleate Boiling Phenomenon Coupled with Thermal Response of the Solid. *Microgravity Sci. Technol.*, 24(4), pp. 255265.

[22] Wu, J., Dhir, V. K., and Qian, J., 2007, Numerical Simulation of Subcooled Nucleate Boiling by Coupling Level-Set Method with Moving-Mesh Method. *Numer. Heat Transf. Part B-Fundam.*, 51(6), pp. 535563.

[23] Wu, J., and Dhir, V. K., 2011, Numerical Simulation of Dynamics and Heat Transfer Associated With a Single Bubble in Subcooled Boiling and in the Presence of Noncondensables. *J. Heat Transf.*, 133(4), pp. 041502-04150214.

[24] Son, G., Ramanujapu, N., and Dhir, V. K., 2002, Numerical Simulation of Bubble Merger Process on a Single Nucleation Site during Pool Nucleate Boiling. *J. Heat Transf.*, 124(1), pp. 5162.

[25] Sato, Y., and Niceno, B., 2015, A Depletable Micro-Layer Model for Nucleate Pool Boiling. *J. Comput. Phys.*, 300, pp. 2052.

[26] Sato, Y., and Niceno, B., 2017, Nucleate Pool Boiling Simulations Using the Interface Tracking Method: Boiling Regime from Discrete Bubble to Vapor Mushroom Region. *Int. J. Heat Mass Transf.*, 105, pp. 505524.

[27] Sato, Y., and Niceno, B., 2018, Pool Boiling Simulation Using an Interface Tracking Method: From Nucleate Boiling to Film Boiling Regime through Critical Heat Flux. *Int. J. Heat Mass Transf.*, 125, pp. 876890.

[28] Son, G., and Dhir, V. K., 2007, A Level Set Method for Analysis of Film Boiling on an Immersed Solid Surface. *Numer. Heat Transf. Part B Fundam.*, 52(2), pp. 153177.

[29] Gibou, F., Fedkiw, R.P., Cheng, L.T. and Kang, M., 2002. A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. *J. Comput. Phys*, 176(1), pp.205-227.

[30] Gibou, F., Chen, L., Nguyen, D., and Banerjee, S., 2007. A Level Set Based Sharp Interface Method for the Multiphase Incompressible NavierStokes Equations with Phase Change. *J. Comput. Phys*, 222(2), pp. 536555.

[31] Aslam, T.D., 2004. A partial differential equation approach to multidimensional extrapolation. *Journal of Computational Physics*, 193(1), pp.349-355.

[32] Tanguy S., Ménard T., Berlemont A., 2007. A level set method for vaporizing two-phase flows. *Journal of Computational Physics*, 221(2), pp. 837-853.

[33] Aktinol, E., and Dhir, V. K., 2014, Numerical Simulation of the Effect of Contact Angle on the Thermal Response of the Solid During Nucleate Boiling. *Interfacial Phenom. Heat Transf.* 2(4), pp. 301-324.

[34] Son, G. and Dhir, V.K., 2008. Numerical simulation of nucleate boiling on a horizontal surface at high heat fluxes. *International Journal of Heat and Mass Transfer*, 51(9-10), pp.2566-2582.

[35] Son, G. and Dhir, V.K., 2008. Three-dimensional simulation of saturated film boiling on a horizontal cylinder. *International Journal of Heat and Mass Transfer*, 51(5-6), pp.1156-1167.

[36] Dhir, V.K., 1991. Nucleate and transition boiling heat transfer under pool and external flow conditions. *International journal of heat and fluid flow*, 12(4), pp.290-314.

[37] Daly, B.J., 1969. A technique for including surface tension effects in hydrodynamic calculations. *Journal of Computational Physics*, 4(1), pp.97-117.

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

[39] Hirt, C.W. and Nichols, B.D., 1981. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of computational physics*, 39(1), pp.201-225.

[40] Welch, S.W. and Wilson, J., 2000. A volume of fluid based method for fluid flows with phase change. *Journal of computational physics*, 160(2), pp.662-682.

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

[42] Gerlach, D., Tomar, G., Biswas, G. and Durst, F., 2006. Comparison of volume-of-fluid methods for surface tension-dominant two-phase flows. *International Journal of Heat and Mass Transfer*, 49(3-4), pp.740-754.

[43] Sun, D.L. and Tao, W.Q., 2010. A coupled volume-of-fluid and level set (VOSET) method for computing incompressible two-phase flows. *International Journal of Heat and Mass Transfer*, 53(4), pp.645-655.

[44] Kunkelmann, C. and Stephan, P., 2010. Numerical simulation of the transient heat transfer during nucleate boiling of refrigerant HFE-7100. *International Journal of Refrigeration*, 33(7), pp.1221-1228.

[45] Wayner Jr, P.C., Kao, Y.K. and LaCroix, L.V., 1976. The interline heat-transfer coefficient of an evaporating wetting film. *International Journal of Heat and Mass Transfer*, 19(5), pp.487-492.

[46] Derjaguin, B.V. and Zorin, Z.M., 1992. Optical study of the adsorption and surface condensation of vapours in the vicinity of saturation on a smooth surface. *Progress in Surface Science*, 40(1-4), pp.83-117.

[47] Kamotani, Y., 1978, January. Evaporator film coefficients of grooved heat pipes. *3rd International Heat Pipe Conference* (p. 404).

[48] Stephan, P.C. and Busse, C.A., 1992. Analysis of the heat transfer coefficient of grooved heat pipe evaporator walls. *International Journal of heat and mass transfer*, 35(2), pp.383-391.

[49] Lay, J.H. and Dhir, V.K., 1995. Shape of a vapor stem during nucleate boiling of saturated liquids. *Journal of Heat Transfer*, 117(2), pp.394-401.

[50] Stephan, P. and Hammer, J., 1994. A new model for nucleate boiling heat transfer. *Heat and Mass Transfer*, 30(2), pp.119-125.

[51] Kunkelmann, C., Ibrahem, K., Schweizer, N., Herbert, S., Stephan, P. and Gambaryan-Roisman, T., 2012. The effect of three-phase contact line speed on local evaporative heat transfer: Experimental and numerical investigations. *International Journal of Heat and Mass Transfer*, 55(7-8), pp.1896-1904.

[52] Wu, J., Dhir, V. K., and Qian, J., 2007, Numerical Simulation of Subcooled Nucleate Boiling by Coupling Level-Set Method with Moving-Mesh Method. *Numer. Heat Transf. Part B-Fundam.*, 51(6), pp. 535563.

[53] Maric, T., Marschall, H., and Bothe, D., 2015, lentFoam - A Hybrid Level Set/Front Tracking Method on Unstructured Meshes. *Comput. Fluids*, 113, pp. 2031.

[54] Nieves-Remacha, M. J., Yang, L., and Jensen, K. F., 2015, OpenFOAM Computational Fluid Dynamic Simulations of Two-Phase Flow and Mass Transfer in an Advanced-Flow Reactor. *Ind. Eng. Chem. Res.*, 54(26), pp. 66496659.

[55] Versteeg, H. K., and Malalasekera, W., 2007, Appendix A- Accuracy of a Flow Simulation. *An Introduction to Computational Fluid Dynamics-The Finite Volume Method*, England.

[56] Patankar, S. V., 1980, Chapter Seven- Finishing Touches. *Numerical Heat Transfer and Fluid Flow*, New York.

[57] Sun, D., Xu, J., and Chen, Q., 2014, Modeling of the Evaporation and Condensation Phase-Change Problems with Fluent. *Numer. Heat Transf. Part B-Fundam.*, 66(4), pp. 326342.

[58] Versteeg, H. K., and Malalasekera, W., 2007, Chapter Five- The Finite Volume Method for Convection-Diffusion Problems. *An Introduction to Computational Fluid Dynamics-The Finite Volume Method*, England.

[59] Kharangate, C. R., and Mudawar, I., 2017, Review of Computational Studies on Boiling and Condensation. *Int. J. Heat Mass Transf.*, 108, pp. 11641196.

[60] Rudman, M., 1997, Volume-Tracking Methods for Interfacial Flow Calculations. *Int. J. Numer. Methods Fluids*, 24(7), pp. 671691.

[61] Son, G., 2003, Efficient Implementation of a Coupled Level-Set and Volume-of-Fluid Method for Three-Dimensional Incompressible Two-Phase Flows. *Numer. Heat Transf. Part B Fundam.*, 43(6), pp. 549565.

[62] Son, G., 2005, A Level Set Method for Incompressible Two-Fluid Flows with Immersed Solid Boundaries. *Numer. Heat Transf. Part B Fundam.*, 47(5), pp. 473489.

[63] Akhtar, M., and Kleis, S., 2012, A Volume Of Fluid Phase Change Model on Adaptive Octree Grids. *Film and Nucleate Boiling Processes*, ASTM International.

[64] Akhtar, M. W., and Kleis, S. J., 2013, Boiling Flow Simulations on Adaptive Octree Grids. *Int. J. Multiph. Flow*, 53, pp. 8899.

[65] Guo, D. Z., Sun, D. L., Li, Z. Y., and Tao, W. Q., 2011, Phase Change Heat Transfer Simulation for Boiling Bubbles Arising from a Vapor Film by the VOSET Method. *Numer. Heat Transf. Part A Appl.*, 59(11), pp. 857881.

[66] Welch, S. W., and Rachidi, T., 2002, Numerical Computation of Film Boiling Including Conjugate Heat Transfer. *Numer. Heat Transf. Part B Fundam.,*, 42(1), pp. 3553.

[67] S. Shipkowski, S.G. Kandlikar,. A Simple to Implement Interface Length Calculation Improvement for Two-phase Fluid Simulations on Square Cells. *To be submitted.*

[68] Brackbill, J., Kothe, D., and Zemach, C., 1992, A Continuum Method for Modeling Surface-Tension. *J. Comput. Phys.*, 100(2), pp. 335354.

[69] Wayner, P. C., 1980, Interfacial Profile in the Contact Line Region of a Finite Contact Angle System. *J. Colloid Interface Sci.*, 77(2), pp. 495500.

[70] Wayner, P. C., 1982, Adsorption and Capillary Condensation at the Contact Line in Change of Phase Heat Transfer. *Int. J. Heat Mass Transf.*, 25(5), pp. 707713.

[71] Kiusalaas, J., 2016, *Numerical Methods in Engineering with MATLAB*, Cambridge University Press.

[72] Stefan, J., 1889, Ueber Die Theorie Der Eisbildung, Insbesondere ber Die Eisbildung Im Polarmeere. *Ann. Phys.*, 278(2), pp. 269286.

[73] Alexiades, V., 1992, *Mathematical Modeling Of Melting And Freezing Processes*, CRC Press, Washington.

[74] Plesset, M. S., and Zwick, S. A., 1954, The Growth of Vapor Bubbles in Superheated Liquids. *J. Appl. Phys.*, 25(4), pp. 493500.

[75] Mikic, B., Rohsenow, W., and Griffith, P., 1970, On Bubble Growth Rates. *Int. J. Heat Mass Transf.* 13(4), pp. 657-.

[76] Scriven, L., 1959, On the Dynamics of Phase Growth. *Chem. Eng. Sci.*, 10(12), pp. 1-.

[77] Tanguy, S., Sagan, M., Lalanne, B., Couderc, F., and Colin, C., 2014, Benchmarks and Numerical Methods for the Simulation of Boiling Flows. *J. Comput. Phys.*, 264, pp. 122.

[78] Sun, D.L., Xu, J.L. and Wang, L., 2012. Development of a vaporliquid phase change model for volume-of-fluid method in FLUENT. *Int. Commun. in Heat and Mass Trans.*, 39(8), pp.1101-1106.

[79] Zhang, L., Li, Z.-D., Li, K., Li, H.-X., and Zhao, J.-F., 2015, Influence of Heater Thermal Capacity on Bubble Dynamics and Heat Transfer in Nucleate Pool Boiling. *Appl. Therm. Eng.*, 88, pp. 118126.

[80] Urbano, A., Tanguy, S., Huber, G., and Colin, C., 2018, Direct Numerical Simulation of Nucleate Boiling in Micro-Layer Regime. *Int. J. Heat Mass Transf.*, 123, pp. 11281137.

[81] Siegel, R., and Keshock, E., 1964, Effects of Reduced Gravity on Nucleate Boiling Bubble Dynamics in Saturated Water. *Aiche J.*, 10(4), pp. 509517.

[82] Hsu, Y.-Y., and Graham, R. W., 1976, *Transport Processes in Boiling and Two-Phase Systems, Including near-Critical Fluids.*, Washington: Hemisphere Pub. Corp.