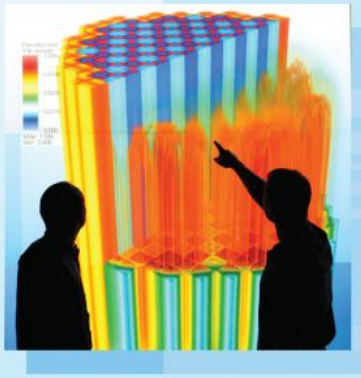
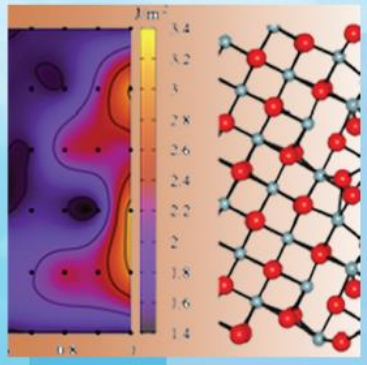
CASL-U-2013-0185-000

Power uprates
and plant life extension



Engineering design
and analysis

Science-enabling
high performance
computing



Fundamental science

Plant operational data

Consortium for Advanced Simulation of LWRs

L3:THM.CLS.P7.09 Advancements on Wall Boiling Modeling in CFD: Leveraging New Understanding from MIT Flow Boiling Facility

Lindsey Gilman, Emilio Baglietto
MIT

Completed: August 30, 2013

U.S. DEPARTMENT OF ENERGY | Nuclear Energy

ADVANCEMENTS ON WALL BOILING MODELING IN CFD: LEVERAGING NEW UNDERSTANDING FROM MIT FLOW BOILING FACILITY

Lindsey Gilman, Emilio Baglietto

August 30, 2013

(Rev. 0)

Massachusetts Institute of Technology
Cambridge, MA, USA
August 2013

CASL-U-2013-0185-000

EXECUTIVE SUMMARY

This milestone introduces advanced multiphase CFD modeling capabilities that are being developed for application to subcooled flow boiling cases. The specific target of this work is to introduce and demonstrate all necessary mechanisms to accurately predict the temperature and heat flux at the wall for subcooled flow boiling. Experiments have illustrated that the current state of the art method, while flexible due to its mechanistic approach, does not account for all the physical phenomena that influence the heat transfer in subcooled flow boiling. The advanced partitioning model is developed using an experimental-based mechanistic approach to accurately capture all the physical phenomena. It also aims to extend the modeling applicability.

The occurrence of sliding bubbles in flow boiling is an important heat transfer contribution that is not being accounted for in current methods. At the MIT flow boiling facility, experiments have illustrated the increased heat transfer caused by sliding bubbles on a heated surface. In many flow boiling conditions, a bubble departs from its nucleation site but remains attached to the heated surface and slides in the direction of the flow until it lifts-off into the bulk flow. While sliding, the bubble disrupts the thermal boundary layer near the wall, which increases the heat transfer while the thermal boundary layer reforms with an influx of cooler liquid on the surface. Additional experiments have also shown the evaporation that occurs through the bubble microlayer and the hot spot on the heated surface that occurs in the dry area of the bubble.

Experimental data such as this has provided strong insight for the heat transfer processes that must be included to accurately model subcooled flow boiling. The proposed model includes:

1) A new evaporation term to truly capture the evaporation occurring on the surface while also tracking the bubble crowding effect on the boiling surface. This includes evaporation from the initial bubble inception and evaporation through the bubble microlayer.

2) A modified convection term to account for increased surface roughness caused by the presence of the bubbles on the heated surface.

3) A new quenching term that accounts for bringing the bubble dry spot back to the wall superheat prior to bubble inception.

4) The addition of a sliding conduction term to capture the increased heat transfer due to bubble sliding on the heated surface prior to lift-off.

Further, improved mechanistic force-balance models for bubble departure and lift-off diameter predictions are implemented in the model. A bubble departure frequency model has also been developed and implemented that is consistent with the bubble departure diameter formulation.

Table of Contents

1 Introduction and Relevance to CASL 5
2 State of the Art Wall Boiling Partitioning 6
2.1 Baseline Boiling Model 6
2.2 Baseline Model Demonstration 7
3 Physical Insight from Experiments 9
3.1 MIT Flow Boiling Facility 9
3.2 Additional Physical Phenomena 11
4 Proposed Model Formulation 14
4.1 Forced Convection Heat Flux 14
4.2 Evaporation Heat Flux 17
4.3 Sliding Conduction 20
4.4 Quenching 23
5 Implemented Submodels 25
5.1 Force Balance Models 25
5.1.1 Bubble Departure Diameter 25
5.1.2 Bubble Lift-Off Diameter 27
5.2 Wall Roughness 27
5.3 Bubble Departure Frequency 27
6 References 29
7 Appendix: Implementation in STAR-CCM+ 33
7.1 Bubble Departure Diameter 33
7.2 Bubble Departure Frequency 47

List of Figures

Figure 1- Bubble departure diameters versus mass flux for 1.05 bar and 15°C subcooling [19]... 9
Figure 2- Nucleation site density for 1.05 bar and 5°C subcooling [19]. 10
Figure 3- High-speed video (top) and IR (bottom) of a bubble from the onset of sliding (at t=0 ms) to the edge of the frame (at t=166 ms) [18]. 10
Figure 4- Velocity of sliding bubbles versus the mass flux [18]. 11
Figure 5- Bubble period versus wall superheat for 1.05 bar and 5°C subcooling [19]. 11
Figure 6- Illustration of a bubble sliding captured at the MIT Flow Boiling Facility by Bren Phillips. Test was completed using water at atmospheric pressure, heat flux of 130 kW/m^2, and 10°C subcooling. 13
Figure 7- Depiction of the heat flux partitioning for subcooled flow boiling. 14
Figure 8- ITM Test Case 1 computational domain and illustration of the hemispherical obstacles [37]. 17
Figure 9- Velocity profile near the wall for a smooth surface and a surface with hemispherical obstacles. 17
Figure 10- Illustration of microlayer evaporation and condensation on a bubble attached to a heated wall. 18
Figure 11- Depiction of the microlayer with the maximum thickness shown. 18
Figure 12 - Illustration of the maximum number of active nucleation sites on a heated surface with a magnified view to show the dry area, microlayer area, and the area of influence for a single bubble. 19
Figure 13- Illustration of the implementation of the maximum active nucleation site density. ... 19
Figure 14- Illustration of bubble growth at a nucleation site until it departs from the site and slides and lifts-off the heater surface. 20
Figure 15- Illustration of a bubble sliding from its inception point and then departing from the heater after sliding a distance $\boldsymbol{l}$. 21
Figure 16- Illustration of bubble growth on a heater and the area in the influence of the bubble that is involved in the quenching heat flux. 24
Figure 17- Bubble schematic with the forces acting on the bubble at a nucleation site [45]. 25

List of Tables

Table 1. Standard Wall Boiling Models 7
Table 2. Definition of variables used in the smooth wall function. 15
Table 3. Forces description and formulation. 26
Table 4. Variable definitions used in the force-balance model. 26

1 Introduction and Relevance to CASL

In many industrial applications, including LWRs, subcooled flow boiling is used when a high heat transfer coefficient is desired. In order to maintain this efficient heat transfer regime, the critical heat flux, which is dependent on the geometry and flow conditions, must be avoided. The need to verify optimal flow conditions for new and enhanced designs has driven studies in numerical analyses, such as CFD, to replace the use of experiments.

To correctly describe critical heat flux using CFD, a complete understanding of the complex boiling phenomena and boiling mechanisms needs to be assembled. A flow boiling facility is available at MIT, where non-invasive techniques are used to capture new and unique subcooled flow boiling data on multiple parameters simultaneously. This facility was specifically constructed to support advanced CFD model development and it provides a complete new look at the boiling phenomenon. The experimental measurements have provided clarification of shortcomings of current model approaches, and most importantly, are providing fundamental data to suggest new thinking that will provide a more general and robust representation.

Comparisons between CFD predictions and experimental data [1], [2], [3] have illustrated the high sensitivity of current boiling closure parameters such as bubble departure diameter, bubble departure frequency, and active nucleation site density on factors such as the wall superheat. Tolubinsky and Kostanchuk [4] predicted the bubble departure diameter by relating it to the degree of local liquid subcooling, which has been used with some success for a limited set of flow parameters. Another model by Kocamustafaogullari [5] suggests that a force balance between gravity and surface tension forces can be used to predict the departure diameter, but this model has been used with limited success in CFD simulations.

The bubble departure frequency model by Cole [6] predicts the frequency by relating it to the typical bubble rise velocity and the bubble departure diameter. More recently proposed bubble departure frequency calculations use the ebullition cycle and account for both the waiting time prior to bubble inception and the time period of bubble growth [7].

The simulation of vapor generation for wall boiling using CFD is extremely sensitive to the active nucleation site density closure parameter. To predict the wide ranges in experimental data for active nucleation site density for a given wall superheat, a dependence on the static contact was added [8] to the model and has been incorporated into the Hibiki and Ishii model [9]. Even with the contact angle dependency, the active nucleation site density models have difficulty capturing the number of active nucleation sites seen experimentally. To remove this dependency on active nucleation site density correlations, a fractal method was developed that treats the active cavity sites analogous to pores in a porous media [10].

The fundamental target of this work is to introduce all necessary mechanisms that must be modeled to accurately predict the temperature and heat flux for subcooled flow boiling at the wall in CFD simulations.

2 State of the Art Wall Boiling Partitioning

The testing and implementation of the wall boiling models is completed using the commercial software STAR-CCM+. The Eulerian MultiPhase (EMP) framework represents the most promising method to model boiling in complex industrial applications and the framework uses the two-fluid, six-equation method to model two-phase flow.

2.1 Baseline Boiling Model

There are various approaches for modeling two-phase flow using CFD. The wall boiling representation in the baseline framework is nevertheless relying on the classic heat partitioning concept introduced by Judd and Hwang (1976) [11] and adapted by Kurul and Podowski (1990) [12] and is shown in 2.1. In the simulation, the total heat flux is computed as the sum of the partitioned components. The convection term describes the removal of heat by single-phase turbulent convection and is shown in 2.2. The quenching term describes the enhancement of heat transfer due to the replacement of a departing bubble by an influx of cooler liquid and is shown in 2.3.

$$q''_{tot} = q''_{fc} + q''_{q} + q''_{e} \qquad 2.1$$

The Del Valle and Kenning model [13] is used to determine quenching heat transfer coefficient and is shown in 2.4. The wait time (t_w) is determined using a wait coefficient that comes from the Kurul and Podowski assumption that quenching occurs between the departure of one bubble and the nucleation of the next. The influence wall area fraction (K_{quench}) also follows the Kurul and Podowski standard model. The evaporation heat flux is shown in 2.5. The bubble departure diameter, active nucleation site density, and frequency for bubble departure are determined using correlations.

$$q''_{fc} = h_{fc}(\Delta T_w + \Delta T_{sub}) \qquad 2.2$$

$$q''_{q} = h_{quench}(\Delta T_w + \Delta T_{sub}) \qquad 2.3$$

$$h_{quench} = 2K_{quench} f \sqrt{\frac{\rho_l c_{pl} k_l t_w}{\pi}} \qquad 2.4$$

$$q''_{e} = \rho_v h_{fg} \frac{\pi}{6} D_d^3 N'' f \qquad 2.5$$

The bulk boiling/condensation is driven by the heat transfer between phases. The heat transfer coefficient for the gas side uses a constant value and the liquid side adopts the Ranz-Marshall correlation [13]. A population balance model can also be implemented to better predict bubble size and distribution in a heated channel.

While the model has shown great flexibility thanks to its detailed mechanistic approach, its ability to correctly predict some of the model parameters in realistic conditions is still challenged. Additionally, some of the fundamental physical characteristics of boiling are not

captured in this partitioning and this can strongly limit the applicability and generality of the approach.

In particular, the movement of bubbles on the heater surface prior to lift-off is not captured in this heat partitioning approach by Kurul and Podowski (1990) since it does not consider the effects of sliding bubbles along the wall [14] which has been shown to occur in subcooled flow boiling through experiments [15], [16]. In subcooled boiling, bubbles often slide along the heated wall after detaching from the nucleation site and before lifting off into the bulk of the liquid flow. These sliding bubbles can also merge and coalesce with other detached and nucleating bubbles downstream. The tendency for bubble sliding is high in subcooled flow boiling and as a result, efforts have been made to incorporate the transient conduction of sliding bubbles in the heat partitioning model [7], [14].

The standard wall boiling models are summarized in Table 1. Although advancements have been made in parameter calculations in subcooled flow boiling modeling using CFD, the need is still present for a modified, more general heat transfer model that uses less sensitive closure parameters and considers the flow effects on boiling.

Table 1. Standard Wall Boiling Models

Parameter	Model
Bubble Departure Frequency	Cole
Bubble Departure Diameter	Tolubinsky-Kostanchuk
Nucleation Site Density	Lemmert Chawla
Bubble Influence Area	Kurul Podowski (2.0)
Quenching Wait Coefficient	Del Valle Kenning (0.8)
Momentum Closure	Symmetric interaction area density
Drag Coefficient	Schiller-Naumann *Lo & Osman [17] suggest Tomiyama
Turbulent Dispersion	$Pr_\tau = 1.0$
Interaction Length Scale	Kurul-Podowski correlation
CHF Relaxation	0.5
CHF Dryout	0.9
Condensation – Fluid	Ranz-Marshall
Condensation- Vapor	Constant $Nu = 2.0$
Interface Temperature	T_{sat}

2.2 Baseline Model Demonstration

To illustrate the capabilities of three-dimensional CFD simulations to accurately model two-phase flow, a study was completed on a 5x5 bundle test from the NUPEC PWR Subchannel and Bundle Test (PBST) International Benchmark exercise by Lo and Osman [17] using the standard wall boiling models and the six-equation, two-fluid model. The rod bundle was modeled with the spacers for the two-phase flow test cases using STAR-CCM+ software. The assessment was completed on the steady-state bundle test B5 Run 5.1121. In this case, the pressure was 167.39 kg/cm^2, the mass flux $14.96x10^6$ kg/m^2hr, the power $2990kW$ (axially uniform), and the inlet temperature 316.9°C.

The CFD results provided good agreement for the void fractions with the experimentally measured data while also providing insight on the detailed flow and void distributions in the rod bundle and how the solid structures affect them. The void fraction was averaged over the central 4 subchannels at a location of $z = 3.177m$ and the experimental value was 0.1791. The computed average over these 4 subchannels was 0.1576 [17]. This shows that current methods can provide reasonable results, but it is important to note that they have been empirically adjusted to do so. The goal of the current work is to provide a more general model that is uses a mechanistic and physics-based approach.

3 Physical Insight from Experiments

3.1 MIT Flow Boiling Facility

A flow boiling facility is available at MIT, where non-invasive techniques (high-speed video and infrared thermography) are used to capture new and unique subcooled flow boiling data on multiple parameters simultaneously [18]. The measurements made include bubble departure diameter, wall superheat (local and surface-averaged), heat transfer coefficient, nucleation site density, and bubble wait time. This facility was specifically constructed to support advanced CFD model development and it provides a complete new look at the boiling phenomenon. The experimental measurements have provided evidence of subcooled flow boiling physical phenomena to suggest new thinking that will provide a more general and robust representation.

The entrance region consists of two channel sections; each with an internal rectangular flow area of 30mm x10mm (300mm^2) and 482.6mm in length. This results in a total entrance region length prior to the quartz cell region of 965.2mm. The quartz cell region has a total length of 220mm and maintains the same rectangular geometry and flow area as the entrance region. The heated area is centered in the quartz cell region and has dimensions of 10mm x 20mm. The heater has a thickness of 0.7µm and is a resistively heated layer of Indium-Tin-Oxide (ITO) using graphite electrodes deposited over a sapphire substrate (1mm). This heater is transparent to IR and therefore a temperature profile can be obtained normal to the heater. The heater is flushed with the wall of the quartz cell so there is no disturbance of the flow over the heated region.

Data collected at pressures of 1.05 and 1.5 bars was completed over a range of mass fluxes, heat fluxes, and subcoolings. It showed that the bubble departure diameter decreased with increasing mass flux and decreasing heat flux, and an example is shown in Figure 1. The nucleation site density increased with increasing wall superheat and decreasing mass flux and is shown in Figure 2. Additionally, localized cooling underneath sliding bubbles was observed. This is shown in Figure 3 for a low heat flux (130kW/m^2/s) and mass flux (200kg/m^2/s) and 10°C subcooling [18], [19].

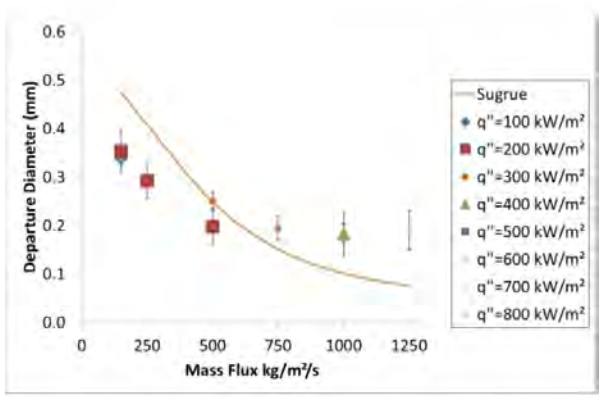


Figure 1- Bubble departure diameters versus mass flux for 1.05 bar and 15°C subcooling [19].

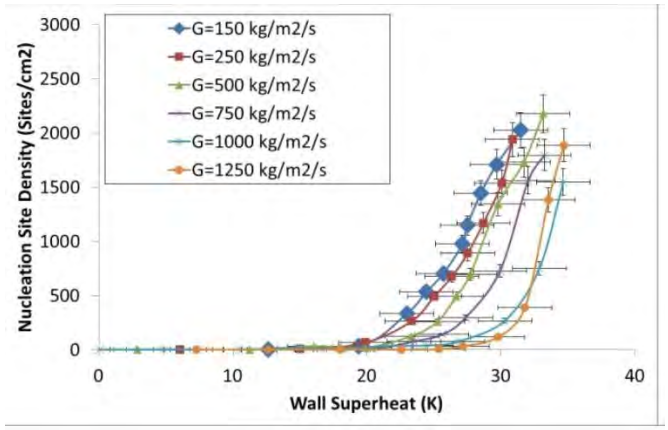


Figure 2- Nucleation site density for 1.05 bar and 5°C subcooling [19].

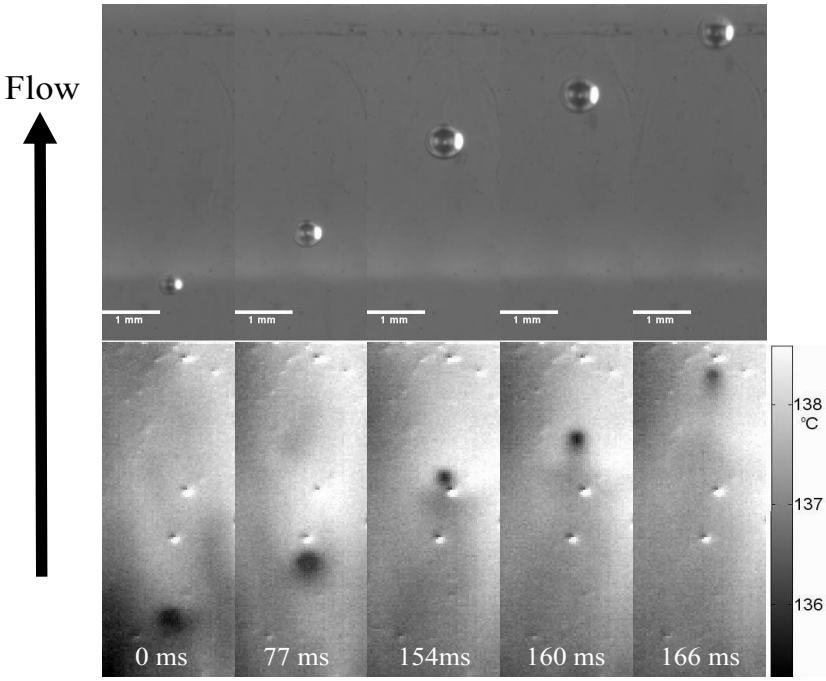


Figure 3- High-speed video (top) and IR (bottom) of a bubble from the onset of sliding (at t=0 ms) to the edge of the frame (at t=166 ms) [18].

The velocity of a sliding bubble was approximately the same speed as the bulk flow at low mass flux and around 50% of the speed of the bulk flow at high mass flux. This is likely caused by the sharp velocity profile near the wall at higher mass flux, where the velocity is about half that of the bulk fluid flow near the wall. The averaged bubble sliding velocities (for approximately 20 bubbles per point) are plotted in Figure 4 as a function of the mass flux. The cooling under the sliding bubbles also tends to be roughly constant from 2-4°C [18].

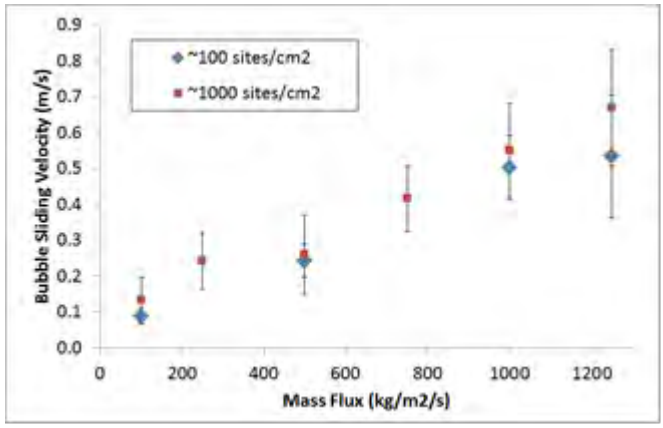


Figure 4- Velocity of sliding bubbles versus the mass flux [18].

The bubble frequency was also investigated by calculating the bubble period using the IR thermography images. This is possible because the nucleation of a bubble is evident by a sharp drop in the wall temperature. Then the wall temperature rises again at the moment of bubble detachment. Figure 5 shows that the bubble period decreases as the wall superheat decreases. The values are compared to the bubble frequency model by Basu [7] and the experimental values consistently have a larger bubble period.

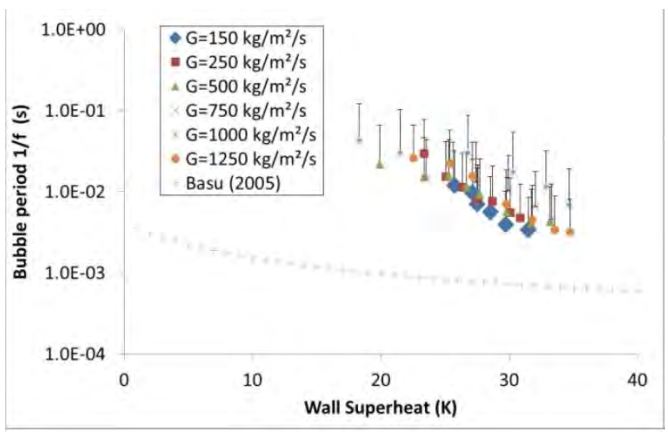


Figure 5- Bubble period versus wall superheat for 1.05 bar and 5°C subcooling [19].

3.2 Additional Physical Phenomena

In subcooled flow boiling, bubbles exhibit different behavior depending on the flow regime. In particular, bubble lift-off diameter from the heater surface is strongly dependent on wall superheat with a slight dependency on flow velocity. Typically the lift-off diameter increases with increasing wall superheat and decreases with increasing velocity. The bubbles also slide at approximately the same velocity as the surrounding liquid [20]. The bubbles can form and remain attached to the heated surface due to the flow conditions that form the thermal boundary. Experiments have shown that the initial bubble growth is rapid and condensation occurs at the top of the bubble [21], [22].

A recent study by Sugrue et al. [23], [24] was conducted to study the bubble departure diameters over a range of pressures, mass fluxes, heat fluxes, subcoolings, and inclination angle of the test section. The investigation indicated that as the inclination angle moves from vertical upflow to horizontal downward-facing orientation, the bubble departure diameter increases.

The importance of upward versus downward flow boiling was highlighted through the high speed camera experimental work conducted by Thorncroft et al. [25]. This study used the fluid FC-87 and slightly subcooled conditions. It showed that in upflow conditions, almost all bubbles slid along the heater wall and did not lift-off immediately. However, in the downflow case, many bubbles lifted-off the heater surface immediately or slid only a small amount on the heater wall prior to lift-off. The upflow experiments had larger heat transfer coefficients than downflow cases with identical operating conditions. This suggests that sliding bubbles have a significant contribution to the overall heat transfer. This study also evaluated the bubble shape while sliding. Bubbles tended to start off spherical and as they grew, became distorted to a cap-like shape. Another study by Li et al. [26] illustrated that the sliding bubble velocities and diameters can vary significantly for a given set of operating conditions. In pool boiling experiments, the influence of active nucleation sites has been shown to lower the bubble departure diameter as sites become closer in proximity [27].

When operating under low heat fluxes, the bubble population is small and the shape of the bubbles is typically spherical. As a bubble grows at a given nucleation site, is also has a slight angle of inclination caused by the asymmetrical bubble growth that occurs in flow boiling [22], [28]. Nearly all the bubbles slide along the heater wall prior to lift-off and change only slightly in size and shape during sliding. The bubbles also typically remain near the wall and, since they remain spherical in shape, and do not travel into the bulk liquid flow. The inclination angle also becomes zero while the bubbles slide on the heater [28], [20]. In this type of flow regime, bubble sliding can be considered the main mode of heat transfer [29], [30]. Both microlayer and transient conduction take place during bubble sliding. The transient conduction is caused by the disruption and reformation of the boundary layer as the bubbles slide [30].

The region of moderate heat fluxes where bubbles grow, detach, and collapse without significantly influencing each other, is referred to as the isolated bubble region. This encompasses most of the region between what is commonly referred to as the onset of nucleate boiling (ONB) and onset of significant voids (OSV). Here, bubbles also slide before lift-off from the heater, but typically only a few diameters or less. In vertical upflow subcooled boiling, larger sized bubbles tend to travel at higher speeds than the surrounding liquid while smaller bubbles travel slightly below the liquid velocity. The bubbles also tend to not be spherical in this region and eject normal from the heated surface into the flow at increased speeds with increasing heat flux and higher subcooling [29].

In this region, the bubbles also tend to change in shape as they slide along the heater wall prior to lift-off. Typically, they start more flat with the longest dimension parallel to the heater surface. Then, just prior to lift-off, the bubbles are elongated in the direction perpendicular to the heater [29]. Other photographic studies of subcooled flow boiling have shown that bubbles can grow significantly while sliding before lift-off [18], [31]. Time snapshots of a growing and sliding bubble are shown in Figure 6.

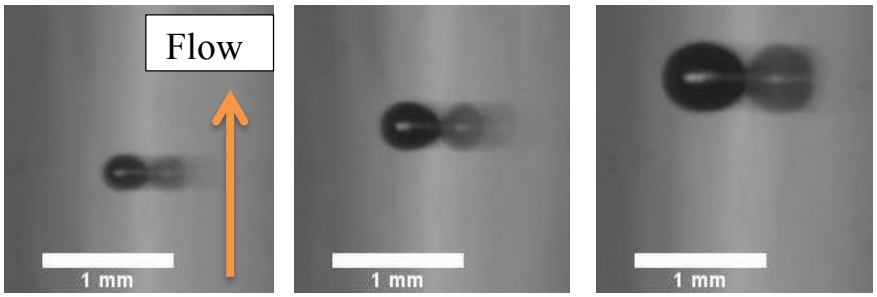


Figure 6- Illustration of a bubble sliding captured at the MIT Flow Boiling Facility by Bren Phillips. Test was completed using water at atmospheric pressure, heat flux of 130 kW/m^2, and 10°C subcooling.

As the heat flux is increased, approaching OSV, the large increase in bubble population allows for interactions between bubbles to occur while on the heater surface. In many cases, bubbles merge prior to detaching from the heater while other bubbles continue to slide and lift-off from the heater unaffected by other bubbles. The interactions between the bubbles can also cause detachment [29]. The heat transfer contribution of sliding bubbles decreases as the heat flux increases because of bubble interaction on the heater [30].

It is known that the nucleation site density increases with wall superheat, but it has also been shown that the activation of new sites at higher temperatures deactivates sites that were active at lower wall superheat [32]. This was noted by Del Valle and Kenning when they studied subcooled flow boiling with water using high speed photography. They also noted the contribution of heat transfer through microlayer evaporation and at high subcoolings, the importance of quenching.

Infrared thermography has also been used with IR transparent heaters, typically made of an optical grade silicon wafer [33]. Since vapor has a very low IR absorptivity, where vapor is in contact with the heated wall the IR camera reads the temperature of the cooler water beyond the vapor. Where the wall is wet, the temperature of the hot water in contact with the wall is measured. This results in IR images that appear dark in dry spots and bright in the wetted area.

The IR images have been used to show the features of a growing bubble. The dry spot is seen in the center of the bubble with a less dark rim which is the microlayer. This has enabled microlayer thickness measurements over time by analysis of the interference fringe patterns created by the IR light passage through the thin microlayer. Additional studies have also noted the importance of microlayer evaporation [22], [34], [35], [36] with the first confirmation of the microlayer by Cooper and Lloyd [36].

At high heat fluxes, this method has shown that even near the critical heat flux, no single point on the surface remains dry due to liquid sloshing on the surface [33]. Additionally at high heat fluxes, the dependence on the surface quantities, such as the nucleation site density, are no longer valid [3]. In the region of OSV, the bubbles have such a high degree of interaction that single bubbles can no longer be tracked [29].

4 Proposed Model Formulation

The fundamental target of this work is to introduce all necessary mechanisms to accurately predict the temperature and heat flux for subcooled flow boiling at the wall in CFD simulations.

The proposed model aims to capture all of the boiling phenomena that occur during subcooled flow boiling. As a result, the wall heat flux is partitioned into four components: forced convection, quenching, evaporation and sliding conduction as is shown in 4.1 and depicted in Figure 7. The addition of the sliding conduction term accounts for the increased heat transfer due to bubbles that slide along the heater wall before lift-off. This is largely due to the transient conduction that occurs from the disrupted thermal boundary layer. These four modes of heat transfer are discussed in detail in the following sections.

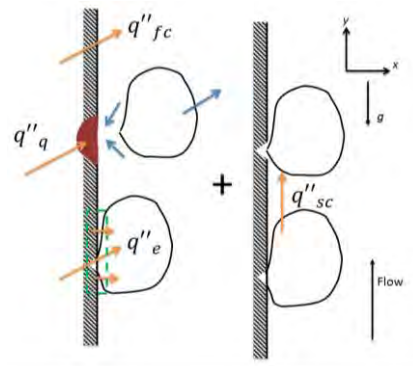


Figure 7- Depiction of the heat flux partitioning for subcooled flow boiling.

$$q''_{tot} = q''_{fc} + q''_{e} + q''_{q} + q''_{sc} \qquad 4.1$$

4.1 Forced Convection Heat Flux

This is calculated using the traditional method of single-phase forced convection on a heated surface. It differs from the base case model in that it also accounts for the addition of increased heat transfer due to the presence of bubbles. This is captured by increasing the surface roughness due to the bubbles on the heater and is done so by modifying the wall function for turbulent flows. The enhanced heat transfer is a function of the size and distribution of bubbles on the heater surface.

For a hydraulically smooth wall, the velocity profile near the wall is given by 4.2 for high-Reynolds number flows. The empirical coefficient (E) is usually set equal to 9.0 and is a constant from the rearrangement of the classic equation (4.3) to give $E = e^{KB}$, where $K =$ 0.42 is the Von Karman constant. The definitions of the remaining variables are listed in

Table 2.

$$u^+ = y^+ \qquad when\ y^+ \leq y_m^+ \qquad 4.2$$

$$u^+ = \frac{1}{K}\ln(Ey^+) \quad when\ y^+ > y_m^+$$

$$u^+ = \frac{1}{K}\ln(y^+) + B \qquad 4.3$$

Table 2. Definition of variables used in the smooth wall function.

Variable	Definition
$\boldsymbol{u^+}$	$\frac{u}{u_\tau}$
$\boldsymbol{u_\tau}$	$\left(\frac{\tau_w}{\rho}\right)^{1/2}$
$\boldsymbol{y^+}$	$\frac{\rho u_\tau y}{\mu}$

To account for the added roughness caused by the bubbles on the heater wall, the rough wall model is employed. This model modifies the log-law coefficient E to make it a function of a roughness parameter given by R^+. This is shown in 4.4, where r is the equivalent sand-grain roughness height. The value for E is then modified by the roughness function, f, such that $E' = E/f$, which is placed in the classic equation for the velocity profile (4.2). The roughness function is dependent on the value of roughness and is shown in 4.5.

$$R^+ = \frac{r u_\tau \rho}{\mu} \qquad 4.4$$

$$f = 1 \qquad when\ R^+ \leq R^+_{smooth} \qquad 4.5$$

$$f = \left[B\left(\frac{R^+ - R^+_{smooth}}{R^+_{rough} - R^+_{smooth}}\right) + CR^+\right]^a \qquad when\ R^+_{smooth} < R^+ < R^+_{rough}$$

$$f = B + CR^+ \qquad when\ R^+ > R^+_{rough}$$

In the intermediate range of wall roughness, where $R^+_{smooth} < R^+ < R^+_{rough}$, a is given by 4.6. Typically, the values for R^+_{smooth}, R^+_{rough}, $B, and\ C$ are 2.25, 90, 0, and 0.253 respectively [37].

$$a = \sin\left[\frac{\pi}{2}\frac{log\left({R^+}/{R^+_{smooth}}\right)}{log\left({R^+_{rough}}/{R^+_{smooth}}\right)}\right] \qquad 4.6$$

To be able to predict the added roughness due to the bubbles, the equivalent sand roughness is estimated and is dependent on both the size and distribution of bubbles on the heater surface. If the bubble sizes are extremely small and lie only in the purely viscous sublayer, when $R^+ \leq R^+_{smooth}$, the wall is treated as a smooth surface because it has no effect on the flow. If the bubbles project further into the flow, when $R^+ > R^+_{rough}$, then it can be treated as fully rough.

Since experimental data is not available to back-calculate the equivalent sand-grain roughness from pressure drop data as is done traditionally [38], the value is estimated by calculating an average surface roughness due to the bubbles. This is shown in 4.7 where n is the number of measurements and y_i is the height for each measurement.

$$R_a = \frac{1}{n}\sum_{i=1}^{n}|y_i| \tag{4.7}$$

The surface of the heater is assumed to be completely smooth except for the area taken up by a bubble. Therefore, everywhere a bubble is not present, the roughness height is zero. A unit cell is used to calculate the average surface roughness. The distance between bubbles is assumed constant and bubbles are distributed in a square lattice so that the bubble spacing is given by $s = 1/\sqrt{N''}$. The average surface roughness is then calculated using an integral form of 4.7 and is shown in 4.8 where $A_{cell} = s^2 = 1/N''$ and V_{tot} is the volume taken up by the single bubble and is dependent on its size and shape. For spherical bubbles, the total volume is calculated by assuming the roughness height is taken at the highest point of the bubble and is shown in 4.9.

$$R_a = \frac{V_{tot}}{A_{cell}} \tag{4.8}$$

$$V_{tot} = {}^{4}/_{3}\,\pi r_b^3\,\frac{1}{2} + \pi r_b^3 = \frac{5}{3}\pi r_b^3 \tag{4.9}$$

An example to illustrate the effectiveness of this method was completed by implementing the results from the ITM-1 study [39]. In this study, the effect of bubbles attached to a wall on both the near-wall turbulence and the friction factor were investigated using the code TransAT. Both Direct Numerical Simulation (DNS) and Large Eddy Simulation (LES) approaches were used to resolve the flow. The computational domain was a Cartesian box of dimensions $L_x = 2\pi h$, $L_y = 2h$, and $L_z = \pi h$ and shown in Figure 8, where h is the half-height of the channel and used as the characteristic lengthscale. The flow is in the +x direction and the domain has periodic boundary conditions in the streamwise (x) and spanwise (z) directions. There is a no-slip boundary at the wall (y). The bubbles were modeled as hemispherical solid obstacles of height k (equivalent to $y+ = 10$) on the walls of the simulation domain. They are spaced at a distance S_b equivalent to $y+ = 40$ and arranged on a square lattice. The simulation was completed using an imposed shear Reynolds number (Re_τ) of 400, a fixed density (ρ) of $1 kg/m^3$, and viscosity (μ) of Re_τ^{-1}.

The average surface roughness was calculated using the integral from described previously. In this test case, the bubbles were modeled as hemispheres, so the $V_{tot} = (2/3)\,\pi r_b^3$. This provided $R_a = 0.00327m$. The rough wall model was implemented in STAR-CCM+ with an equivalent sand grain roughness value given by the average surface roughness value calculated. This is compared to the DNS data for a smooth wall case and the hemispherical case to illustrate the ability of this method to capture the roughness effects on the velocity profile of the flow and is shown in Figure 9. This is important for forced convective heat transfer because the temperature profile is then obtained from this velocity profile.

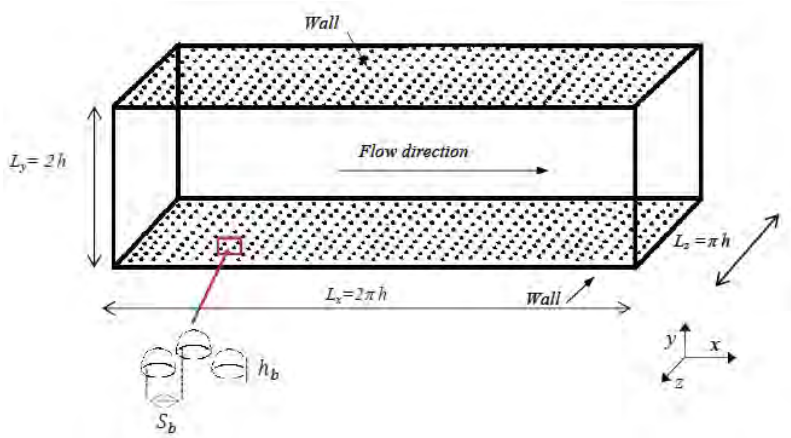


Figure 8- ITM Test Case 1 computational domain and illustration of the hemispherical obstacles [39].

The single-phase forced convection component is employed on all areas of the heater that is not under bubble influence. The calculation for the forced convection component is given by 4.10, where h_{fc} is the heat transfer coefficient that incorporates the increased heat transfer due to the presence of bubbles when wall boiling occurs.

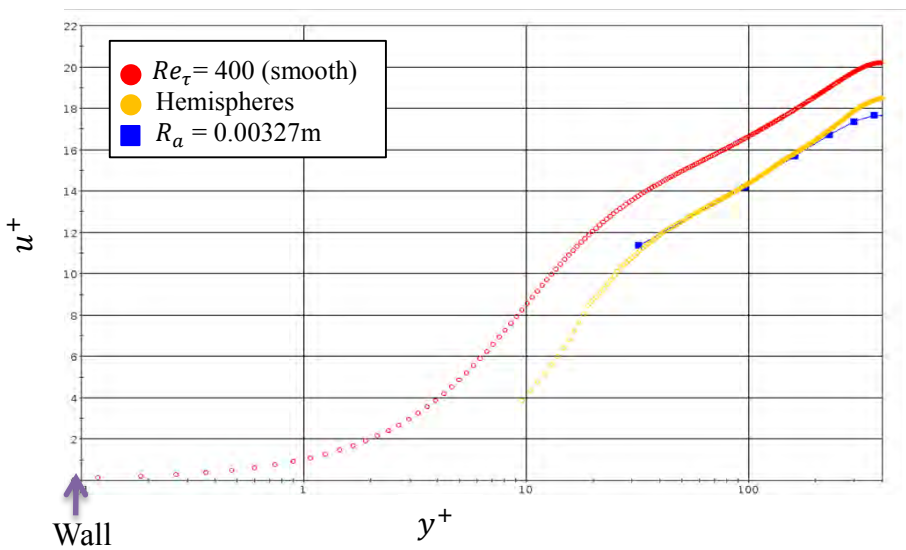


Figure 9- Velocity profile near the wall for a smooth surface and a surface with hemispherical obstacles.

$$q''_{fc} = h_{fc}(\Delta T_w + \Delta T_{sub}) \qquad 4.10$$

4.2 Evaporation Heat Flux

The evaporation term is calculated using the physical phenomena of the rapid initial bubble growth and the microlayer evaporation and is shown in 4.11. The initial bubble growth is calculated as shown in 4.12. The microlayer is a thin layer of liquid that becomes trapped between a quickly growing bubble and the superheated wall and is illustrated in Figure 10.

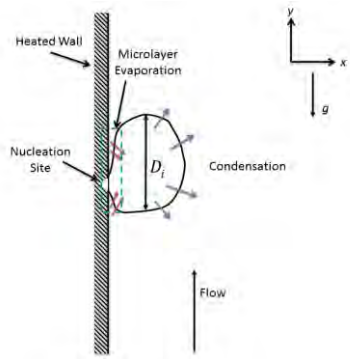


Figure 10- Illustration of microlayer evaporation and condensation on a bubble attached to a heated wall.

$$q''_e = q''_{e,init} + q''_{e,mL} \tag{4.11}$$

$$q''_{e,init} = V_{b,final}\rho_v h_{fg} f N'' \tag{4.12}$$

The volume of the microlayer is illustrated in Figure 11 and is calculated by assuming the microlayer has a maximum thickness (δ_{max}) that decreases to zero at the center of the bubble. It is assumed that the entire microlayer evaporates for each bubble and is given in 4.13 and 4.14. The value used for the radius of the microlayer is half the radius of the bubble and $\delta_{max} = 2\mu m$.

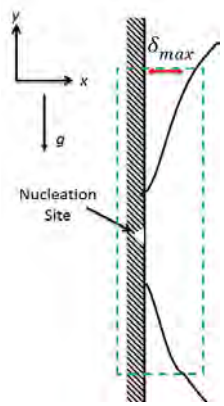


Figure 11- Depiction of the microlayer with the maximum thickness shown.

$$q''_{e,mL} = V_{mL}\rho_f h_{fg} f N'' \tag{4.13}$$

$$V_{mL} = \frac{2}{3}\pi\left(\frac{R}{2}\right)^2 \delta_{max} \tag{4.14}$$

Literature sources vary on the reported size of the influence area factor [11], [12] which consequently affects the number of bubbles that can fit on the heater surface. With changing operating conditions, preliminary experimental data shows that the influence area of the bubble stays relatively constant so the value is currently set as $F_A = 2$ [18]. The area of influence term takes into account that the bubble on the heater surface affects an area larger than the physical

area of the bubble. It must also account for "crowding," which is when the number of bubbles on the heater surface begin to affect each other [32]. Since the current active nucleation site density models are highly dependent on the wall superheat and consequently can reach unrealistic values, a maximum value is enforced that is determined on the physical number of bubbles that can fit on the heater surface. On average, this maximum number is depicted in Figure 12 along with a magnified view showing the dry area, microlayer area, and the area of influence for a single bubble. Through this depiction, the effective area of a single bubble is calculated as is shown in 4.15, where D_m is the maximum bubble size reached before the bubble departs from the nucleation site.

$$a_{b,eff} = F_A \pi \left(\frac{D_m}{2}\right)^2 \qquad 4.15$$

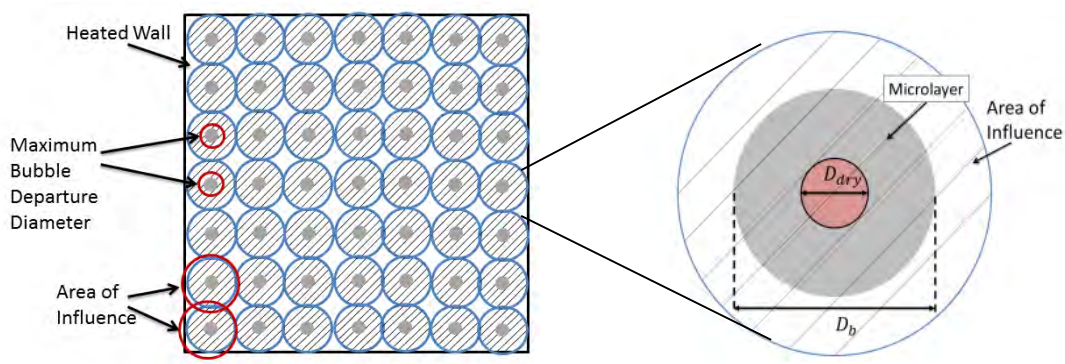


Figure 12 - Illustration of the maximum number of active nucleation sites on a heated surface with a magnified view to show the dry area, microlayer area, and the area of influence for a single bubble.

The effective area of the bubbles is used to suppress the drastic increase in active nucleation site density at high wall superheat to prevent unphysical values. Current model implementation is depicted in Figure 13 where the maximum active nucleation site density is given by 4.16.

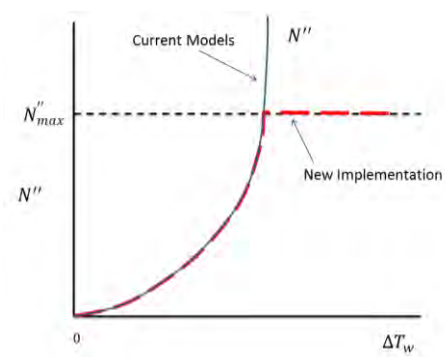


Figure 13- Illustration of the implementation of the maximum active nucleation site density.

$$N''_{max} = \frac{1}{a_{b,eff}} \qquad 4.16$$

4.3 Sliding Conduction

As experiments have illustrated, the effect of sliding bubbles on the heat transfer coefficient for subcooled flow boiling can be very high, even the dominant mode of heat transfer for particular flow regimes.

A mechanistic force balance model is used to predict if a bubble will slide along the heated wall, in the same fashion as it is currently used to predict when the bubble will lift-off the surface. The original concept was proposed for both pool and flow boiling by Klausner et al. [41] and Zeng et al. [20] and it was recently adapted and modified by Yun et al. [42] for CFD application.

The bubble departure diameter, or bubble movement diameter (D_m), is defined as the diameter of the bubble when it moves from its inception point, or point of origin. This diameter is predicted by the force balance model by Sugrue et al. [43], which is a modified version of the version developed by Yun et al. and described in detail in Section 5.1. At this size, the bubble may slide along the heater surface and lift-off when it reaches the lift-off diameter size (D_l). The lift-off diameter is also predicted by a force-balance model developed by Situ et al. [22] and is illustrated in Figure 14. If the prediction of D_l is smaller than D_m, then the bubble lifts off into the fluid flow without sliding.

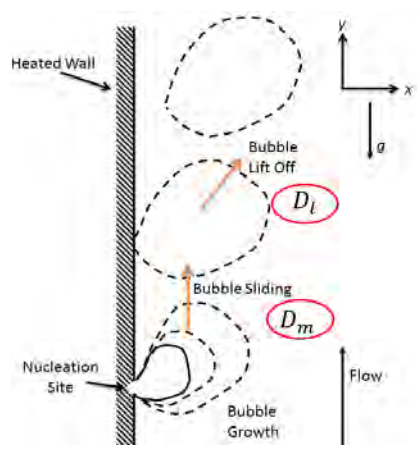


Figure 14- Illustration of bubble growth at a nucleation site until it departs from the site and slides and lifts-off the heater surface.

Bubble sliding increases heat transfer because of the disruption of the thermal boundary layer inducing transient conduction as the cold liquid comes in contact with the wall. This is modelled using the error function solution to the transient temperature profile for a 1-D transient heat conduction into a semi-infinite medium using the heater surface temperature as T_w and the liquid temperature as T_l [7] and the wall heat flux is shown in 4.17. When the boundary layer is disrupted by a sliding bubble, the sliding conduction heat transfer mode occurs for a specific time interval while the thermal boundary layer is re-established which is given by t^*. This time interval is determined by the fraction of time the transient conduction heat transfer coefficient is

greater than the forced convection term and is given in 4.18. By integrating the error function solution over this time, the sliding conduction term is given by 4.19.

$$q_w'' = \frac{k_l(T_w - T_l)}{\sqrt{\pi \alpha_l t}} \tag{4.17}$$

$$t^* = \left(\frac{k_l}{h_{fc}}\right)^2 \frac{1}{\pi \alpha_l} \tag{4.18}$$

$$q_{sc}'' = \frac{k_l(T_w - T_l)}{\sqrt{\pi \alpha_l t^*}} f a_{sl} t^* N'' \tag{4.19}$$

The sliding length and area influenced by the sliding bubbles (a_{sl}) is determined by the bubble growth while sliding and also the number of additional nucleation sites it may cross while sliding. The bubble is assumed to slide at the velocity of the liquid in its proximity and only in the direction of the flow. As in the forced convection term, the spacing between the bubbles, s, is calculated by assuming the bubbles are arranged in a square lattice arrangement. The distance a bubble slides is depicted in Figure 15, where l is the total distance a bubble slides. The calculation of l is dependent on whether the bubble slides a distance greater or less than the spacing s.

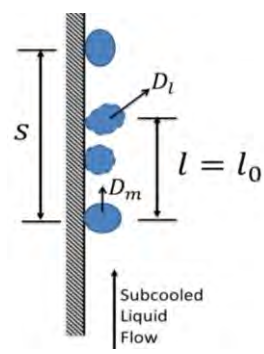


Figure 15- Illustration of a bubble sliding from its inception point and then departing from the heater after sliding a distance *l*.

The growth of the bubble while sliding is determined from a correlation against Maity's data [44] of a single bubble sliding in subcooled flow boiling and is given in 4.20 where D_{sl} is the bubble diameter after sliding a time t_{sl} and D_{in} is the diameter of the bubble when it begins to slide. Re_b is the Reynolds number of the bubble and Ja is the Jakob number given in 4.21 and 4.22 respectively. U is the liquid velocity at the center of the bubble and ν_l is the kinematic viscosity of the liquid.

$$\frac{(D_{sl}^2 - D_{in}^2)}{t_{sl}\alpha_l Ja_{sup}} = \frac{1}{15(0.015 + 0.0023 Re_b^{0.5})(0.04 + 0.023 Ja_{sub}^{0.5})} \tag{4.20}$$

$$Re_b = \frac{2UR}{\nu_l} \tag{4.21}$$

$$Ja = \frac{\rho_l c_{pl} \Delta T}{\rho_v h_{fg}} \quad 4.22$$

For example, if a bubble departs from its inception point and slides a distance s, then the time it takes to slide is given by 4.24 and $D_{in} = D_m$. The size after it slides (D_{sl}) can then be calculated. If the diameter after it slides is less than the lift-off diameter, then it will continue to slide and it also "absorbs" the bubble that was at the second nucleation site that the original bubble encountered. The addition of the volume of the bubble swept by the original sliding bubble is given by 4.23. Since most of the ebullition cycle is the wait time rather than the growth time, the additional volume added to the sweeping bubble is a fraction of the bubble departure diameter that uses the ratio of the growth time versus total time for an ebullition cycle.

$$D_{add} = t_g f D_m \quad 4.23$$

The total distance a bubble slides can then be calculated as shown in 4.25 where N_{merged} is the number of bubbles the sliding bubble merged with on the heater and $l_{D_n - D_l}$ is the additional length the sliding bubble travelled from the last bubble merger to lift-off. Using the average bubble size during sliding, as given in 4.26, the area affected by a sliding bubble is then $a_{sl} = D_{avg} l$.

$$t_{sl} = \frac{s}{v_b} \quad 4.24$$

$$l = N_{merged} s + l_{D_n - D_l} \quad 4.25$$

$$D_{avg} = D_m + 0.5(D_l - D_m) \quad 4.26$$

To account for the bubbles that are "absorbed" by the sliding bubble as it passes over other active nucleation sites, the active nucleation site density is reduced. This is completed by employing a reduction factor when the sliding distance is greater than the spacing between bubbles that is given by 4.27 [7]. This provides a new active nucleation site density value that is shown in 4.28.

$$R_f = \frac{1}{l/s} = \frac{1}{l\sqrt{N''}} \quad 4.27$$

$$N''^* = R_f N'' \quad 4.28$$

The bubble departure and lift-off diameter force-balance models use a strong assumption that the departing bubbles are approximately spherical in shape. Once the bubbles begin to deviate from the spherical shape, the forces acting on the bubble can dramatically change. Most importantly is the shear lift force that causes more deformed bubbles to move away from the walls and into the bulk flow. To account for the effect of bubble deformability in bubble lift-off, the Eötvös (Eo) number is used and is shown in 4.29 where D is the bubble diameter and σ is the surface tension.

The Eo is proportional to the ratio of the buoyancy force to the surface tension force. Therefore, at low Eo numbers, the surface tension is sufficient enough for the bubble to remain spherical in shape. Typically, the Eo number is also described with either the Morton (Mo) or Galileo (N) number to characterize the bubble shape in the surrounding fluid. The Mo number is a constant for a given fluid and given in 4.30. For low viscosity liquids, the Mo number is lower [45], [46].

$$Eo = \frac{(\rho_l - \rho_v) g D^2}{\sigma} \tag{4.29}$$

$$Mo = \frac{(\rho_l - \rho_v) g \mu_l^4}{\rho_l^2 \sigma^3} \tag{4.30}$$

Since water is a low viscosity fluid, the Mo number is low. Therefore, when the Eo number is also low (for smaller bubbles), the bubbles are spherical in shape. As the Eo number increases, the bubbles become ellipsoidal in shape and "wobbly." They can eventually have a spherical cap shape if the Eo number becomes large enough. For high-Renyolds number flows, if the Eo number is at or below 0.1, the bubbles are essentially spherical [45], [46].

A DNS study still under progress by Dabiri et al. investigated the regime transitions in vertical channel upflow due to bubble deformability. The transition from low to high flow rate occurs because of the location of the bubbles in the channel At low Eo, the bubbles are spherical and stay near the walls causing a lower flow rate than the single-phase solution. As Eo rises, the bubbles move to a more uniform distribution and the flow rate increases to the single-phase solution. This transition occurs rather abruptly at an Eo number of 2.5 and when Eo is greater than 3.5, there are no bubbles sliding along the walls. Therefore, when the Eo number is greater than 3.5, sliding bubbles are no longer considered as a mode of heat transfer.

4.4 Quenching

The quenching component in this new model is governed by the heated material because this heat transfer mode involves bringing the dry area of the bubble on the heater back to the wall superheat and temperature distribution prior bubble inception. In this method, the quenching term becomes a function of the material properties of the heated surface. It is employed when a bubble departs from its nucleation site and is depicted in Figure 16. The quenching heat flux is shown in 4.31 where ρ_h is the density of the heater material, $c_{p,h}$ is the heat capacity, ΔT is the average temperature difference between the hot spot on the heater and the surrounding wall temperature in the bubble location (approximated as 2K), and V_q is the volume of the hot spot and given in 4.32 where volume is assumed to be a hemisphere of the same diameter as the dry area. The dry area is assumed to have half the radius of the bubble.

$$q_q'' = \rho_h c_{p,h} \Delta T V_q f N'' \tag{4.31}$$

$$V_q = \frac{4}{3}\pi \left(\frac{D_{dry}}{2}\right)^3 \frac{1}{2} = \frac{2}{3}\pi \left(\frac{D_{dry}}{2}\right)^3 \tag{4.32}$$

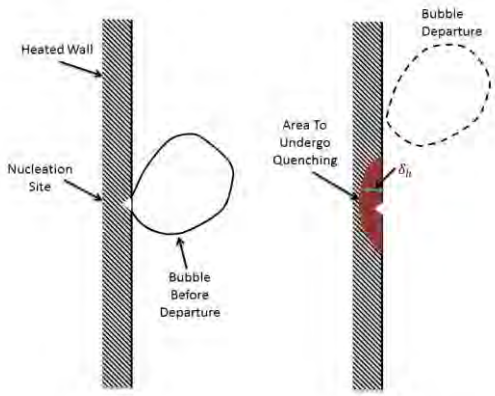


Figure 16- Illustration of bubble growth on a heater and the area in the influence of the bubble that is involved in the quenching heat flux.

5 Implemented Submodels

5.1 Force Balance Models

As was described in Section 4, many of the calculations for the heat flux partitioning are dependent on the bubble departure and lift-off diameters. Therefore, these values are of high importance for the model. As a result, a mechanistic force balance model by Sugrue et al. [43] is implemented in the code that modifies the original force-balance by Klausner et al. [41] and Zeng et al. [20] and the CFD implemented version by Yun et al. [42]. This revised model was built by systematically investigating experimental data of bubble departure diameters over a large range of mass fluxes, test section angle orientations, pressures, and subcoolings. Another force-balance model proposed by Situ et al. [22] and tested also by [28] incorporates the physical phenomena of a sliding bubble to better predict the lift-off diameter.

5.1.1 Bubble Departure Diameter

The mechanistic bubble departure model by Sugrue et al. [43] for flow boiling employs a summation of all the forces acting both parallel and perpendicular to the heater on which the bubble forms. Once the summation of these forces is greater than zero, the bubble either lifts-off the heater (net force greater than zero perpendicular to heater) or slides (net force greater than zero tangential to heater). Figure 17 illustrates the forces calculated. The formulation of these forces is shown in Table 3. The summation of these forces is calculated with a dependence on the orientation of the heater and is shown in 5.1 and 5.2. As shown, ϑ is the orientation angle of the heater surface, and ϕ is the inclination angle representing the direction of bubble growth with respect to the y-axis (a constant of $\pi/18$ is implemented). Additional variable definitions are given in Table 4.

This model is employed to predict the bubble diameter when it departs from its nucleation site. If it is predicted to lift-off the heater, then no sliding occurs. If the bubble is predicted to depart in the direction of the flow, it slides along the heater surface until it reaches the lift-off diameter predicted in the following section.

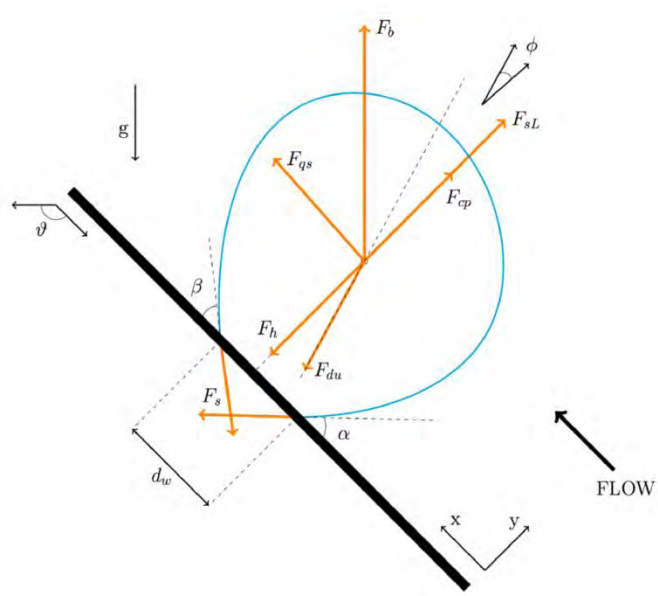


Figure 17- Bubble schematic with the forces acting on the bubble at a nucleation site.

Table 3. Forces description and formulation [43].

Abbreviation	Name	Formulation
F_{sx}	Surface Tension (x-direction)	$F_{sx} = -1.25 d_w \sigma \frac{\pi(\alpha - \beta)}{\pi^2 - (\alpha - \beta)^2}[\sin\alpha + \sin\beta]$
F_{sy}	Surface Tension (y-direction)	$F_{sy} = -d_w \sigma \frac{\pi}{\alpha - \beta}[\cos\beta - \cos\alpha]$ $d_w = 0.025 d_b$
F_{du}	Unsteady Drag	$F_{du} = -\rho_l \pi R^2 \left(\frac{3}{2} C_s \dot{R}^2 + R\ddot{R}\right); \quad C_s = 1$ $R(t) = \frac{2b}{\sqrt{\pi}} Ja\sqrt{\eta t}$
F_b	Buoyancy	$F_b = V_b(\rho_l - \rho_v)g = \frac{4}{3}\pi R^3(\rho_l - \rho_v)g$
F_{qs}	Quasi-Steady Drag	$F_{qs} = 6\pi n \mu_l \Delta v R\left[\frac{2}{3} + \left[\left(\frac{12}{\mathrm{Re}_b}\right)^n + 0.796^n\right]^{-1/n}\right]$
F_{sL}	Shear Lift	$F_{sL} = \frac{1}{2}\rho_l \Delta v^2 \pi R^2\left[3.877 G_s^{1/2}\left[\mathrm{Re}_b^{-2} + 0.014 G_s^2\right]^{1/4}\right]$
F_h	Hydrodynamic Pressure	$F_h = \frac{9}{8}\rho_l \Delta v^2 \frac{\pi d_w^2}{4}$
F_{cp}	Contact Pressure	$F_{cp} = \frac{\pi d_w^2}{4}\frac{2\sigma}{5R}$

$$\sum F_x = F_{sx} + F_{qs} + F_b \sin\theta + F_{du}\sin\phi \qquad 5.1$$

$$\sum F_y = F_{sy} + F_{sL} - F_b\cos\theta - F_h + F_{cp} + F_{du}\cos\phi \qquad 5.2$$

Table 4. Variable definitions used in the force-balance model.

α	Advancing contact angle	Ja	Jakob number
β	Receding contact angle	b	Aspherity constant (1.56)
d_w	Bubble foot diameter	g	Gravity
σ	Surface tension	ν	Kinematic viscosity
R	Instantaneous Bubble radius	n	Constant (0.65)
d_b	Bubble diameter	Re_b	Reynolds number of bubble
η	Thermal diffusivity	G_s	Dimensionless shear rate of flow
t	time	Δv	Flow velocity (U)

$$Re_b = \frac{2UR}{\nu} \qquad 5.3$$

$$G_s = \left|\frac{dU}{dy}\right|\frac{R}{U} \qquad 5.4$$

$$u^* = \sqrt{\frac{\tau_w}{\rho}} \qquad 5.5$$

$$Ja = \frac{\rho_l c_{pl}\Delta T_{sat}}{\rho_v h_{fg}} \qquad 5.6$$

$$\frac{U(y)}{u^*} = \frac{1}{\kappa} ln(1 + \kappa y^+) + c\left[1 - exp\left(-\frac{y^+}{\chi}\right) - \frac{y^+}{\chi} exp(-0.33y^+)\right] \qquad 5.7$$

5.1.2 Bubble Lift-Off Diameter

When a bubble is sliding, the forces acting on the bubble are different from the model by Sugrue et al. where the bubble is static on the heater surface. At the moment of lift-off, the surface tension, hydrodynamic, and contact pressure forces for the bubble can be neglected because the bubble foot diameter, or bubble contact diameter on the heater surface, is essentially zero. Additionally, it has been shown that the inclination angle goes to zero when the bubble slides, so the unsteady growth (drag) force is normal to the flow direction. This leaves two or three competing forces to predict the lift-off diameter depending on the orientation of the heater (the buoyancy force is dependent on test section orientation) [22], [28].

F_{sL}^{*} is the shear lift-force that is different from that calculated for the static bubble because the relative velocity between the bubble and surrounding fluid is no longer simply the liquid velocity at the center of the bubble. The relative velocity is calculated assuming the bubble flows at the same velocity as the fluid near the wall. Then the relative velocity is the difference between the velocity near the wall (at a distance of y=R) and the liquid velocity a bubble diameter away. The force balance lift-off formulation is also similar to an approach developed by Zeng et al. [20] for horizontal flow boiling. A force balance analysis by Yeoh and Tu [10] also showed that the bubble is governed by the growth force and hear lift force at the instant of lift-off. They also developed a dimensionless bubble lift-off diameter model as a function of the effective Jakob number and the Prandtl number.

$$\sum F_y = F_{du} + F_{sL}^{*} - F_b cos\vartheta \qquad 5.8$$

5.2 Wall Roughness

As was descried in Section 4.1, the bubble influence on the wall roughness is captured by using the rough wall model for the velocity profile in high-Reynolds number flows. The bubbles are assumed to be spherical in shape and distributed in a square lattice on the surface of the heater. This provides the integral form for the average surface roughness as shown in 5.9. This average surface roughness value is implemented through the equivalent sand-grain roughness height for the rough wall model.

$$R_a = \frac{5}{3}\pi r_b^3 N'' \qquad 5.9$$

5.3 Bubble Departure Frequency

The bubble departure frequency is determined by employing the bubble ebullition cycle. This refers to both the bubble wait time, the time between one bubble departure and the nucleation of the next bubble, and the growth time, the time from the bubble nucleation to departure from the nucleation site. The wait and growth times added together give the bubble period and the inverse of which is the bubble departure frequency as given in 5.10.

$$f = \frac{1}{t_w + t_g} \tag{5.10}$$

The growth time is calculated using the same bubble growth equation that is used in the force-balance bubble departure model. This uses Zuber's model [47] which is the same model used by Yun [42] but neglecting the component that accounts for subcooling. When the bubble departure diameter, R, is known through the force-balance prediction, the growth time is calculated as shown in.

$$t_g = \frac{\pi R^2}{4b^2 Ja^2 \eta_l} \tag{5.11}$$

The wait time is calculated using the fractal model developed by Yeoh and Tu [10]. This method assumes a critical cavity radius (R_c) for a nucleation site to be active and is a function of the wall superheat. The calculation for the critical diameter is given in 5.12 where ΔT_{sup} is the wall superheat. The wait time is then calculated as shown in 5.13 where $C_1 = (1 + cos\theta)/sin\theta$ and $C_2 = 1/sin\theta$ and θ is the bubble contact angle.

$$D_c = \frac{4\sigma T_{sat}}{\rho_v h_{fg} \Delta T_{sup}} \tag{5.12}$$

$$t_w = \frac{1}{\pi\alpha}\left[\frac{(T_w - T_l)R_c C_1}{(T_w - T_{sat}) - 2\sigma T_{sat}/C_2 R_c \rho_v h_{fg}}\right] \tag{5.13}$$

This formulation is quite different from the more traditionally used correlation by Cole (1960) that is highly dependent on the bubble size [6].

6 References

[1] C. Gerardi, J. Buongiorno, L.-W. Hu and T. McKrell, "Infrared Thermometry Study of Nanofluid Pool Boiling Phenomena," *Nanoscale Research Letters,* vol. 6, no. 232, 2011.

[2] J. Garnier, E. Manon and G. Cubizolles, "Local Measurements on Flow Boiling of Refrigerant 12 in a Vertical Tube," *Multiphase Science and Technology,* vol. 13, pp. 1-111, 2001.

[3] T. Theofanous and T. Dinh, "High Heat Flux Boiling and Burnout as Microphysical Phenomena: Mounting Evidence and Opportunities," *Multiphase Science and Technology,* vol. 18, no. 3, pp. 251-276, 2006.

[4] V. Tolubinsky and D. Kostanchuk, "Vapour bubbles growth rate and heat transfer intensity at subcooled water boiling," in *Heat Transfer 1970, Preprints of papers presented at the 4th International Heat Transfer Conference*, Paris, 1970.

[5] G. Kocamustafaogullari, "Pressure dependence of bubble departure diameter for water," *Int. Comm. Heat Mass Transfer,* vol. 10, pp. 504-509, 1983.

[6] R. Cole, "A photographic study of pool boiling in the region of the critical heat flux," *AIChE J.,* vol. 6, pp. 533-542, 1960.

[7] N. Basu, G. R. Warrier and V. K. Dhir, "Wall Heat Flux Partitioning During Subcooled Flow Boiling: Part 1- Model Development," *Journal of Heat Transfer,* vol. 127, no. 2, pp. 131-140, 2005.

[8] N. I. Kolev, "How Accurately can We Predict Nucleate Boiling?," *Experimental Thermal and Fluid Science,* vol. 10, no. 3, pp. 370-378, 1995.

[9] T. Hibiki and M. Ishii, "Active nucleation site number density in boiling systems," *Int. J. Heat Mass Transfer,* vol. 46, pp. 2587-2601, 2003.

[10] G. Yeoh, S. Cheung, J. Tu and M. Ho, "Modeling Vertical Subcooled Boiling Flows at Low Pressures," *Journal of ASTM International,* vol. 8, pp. 1-20, 2011.

[11] R. Judd and K. Hwang, "A comprehensive model for nucleate boiling heat transfer including microlayer evaporation," *J Heat Transfer,* vol. 115, pp. 955-962, 1976.

[12] N. Kurul and M. Podowski, "Multidimensional effects in sub-cooled boiling," in *Proceedings of the Ninth Heat Transfer Conference*, Jerusalem, 1990.

[13] W. E. Ranz and W. Marshall, "Evaporation from drops--Part I and II," *Chemical Engineering Progress,* vol. 48, p. 141, 1952.

[14] G. H. Yeoh and J. Tu, Computational Techniques for Multi-Phase Flows, Elsevier, 2010.

[15] M.-O. Kim, J. Tu, G. H. Yeoh and G.-C. Park, "On Population Balance Approach for Subcooled Boiling Flow Prediction," *Journal of Heat Transfer,* vol. 127, pp. 253-264, 2005.

[16] T. Lee, G. Park and D. Lee, "Local Flow characteristics of Subcooled Boiling Flow of Water in a Vertical Concentric Annulus," *International Journal of Multiphase Flow,* vol. 28, no. 8, pp. 1351-2368, 2002.

[17] S. Lo and J. Osman, "CFD Modeling of Boiling Flow in PSBT 5x5 Bundle," *Hindawi Publishing Corporation,* Vols. 2012, Article ID 795935, p. 8, 2012.

[18] B. Phillips, J. Buongiorno and T. McKrell, "Nucleation Site Density, Bubble Departure Diameter, Wait Time and Local Temperature Distribution in Subcooled Flow Boiling of Water at Atmospheric Pressure," in *NURETH-15*, Pisa, Italy, May 12-17, 2013.

[19] B. Phillips, J. Buongiorno and T. McKrell, "Nucleation Site Density, Bubble Departure Diameter, Bubble Departure Freqency, and Local Temperature Distribution in Subcooled Flow Boiling of Water," CASL L3.THM.CLS.P7.01, Cambridge, MA, 2013.

[20] L. Zeng, J. Klausner, D. Bernhard and R. Mei, "A Unified Model for the Prediction of Bubble Detachment Diameters in Boiling Systems- II. Flow Boiling," *International Journal of Heat and Mass Transfer,* vol. 36, no. 9, pp. 2271-2279, 1993.

[21] M. Salcudean and E. Bibeau, "A Study of Bubble Ebullition in Forced-Convection Subcooled Nucleate Boiling at Low Pressure," *International Journal of Heat and Mass Transfer,* vol. 37, no. 15, pp. 2245-2259, 1994.

[22] R. Situ, T. Hibiki, M. Ishii and M. Mori, "Bubble Lift-Off Size in Forced Convective Subcooled Boiling Flow," *International Journal of Heat and Mass Trasnfer,* vol. 48, pp. 5536-5548, 2005.

[23] R. Sugrue, The Effects of Orientation Angle, Subcooling, Heat Flux, Mass Flux, and Pressure on Bubble Growth and Detachment in Subcooled Flow Boiling, Cambridge, MA: Massachusetts Institute of Technology, 2012.

[24] R. Sugrue, T. McKrell and J. Buongiorno, "On the Effects of Orientation Angle, Subcooling, Mass Flux, Heat Flux, and Pressure on Bubble Departure Diameter in Subcooled Flow Boiling," in *CFD4NRS*, 2013.

[25] G. Thorncroft, J. Klausner and R. Mei, "An experimental investigation of bubble growth and detachment in vertical upflow and downflow boiling," *International Journal of Heat and Mass Transfer,* vol. 41, pp. 2857-3871, 1998.

[26] S. Li, S. Tan, C. Xu, P. Gao and L. Sun, "An experimental study of bubble sliding characteristics in narrow channel," *International Journal of Heat and Mass Transfer,* vol. 55, pp. 488-497, 2012.

[27] C. Hutter, K. Sefiane, T. Karayiannis, A. Walton, R. Nelson and D. Kenning, "Nucleation site interaction between artificial cavities during nucleate pool boiling on silicon with integrated micro-heater and temperature micro-sensors," *International Journal of Heat and*

Mass Transfer, vol. 55, pp. 2769-2778, 2012.

[28] R. H. Khanlou, A. Mohammadi and S. A. Jazayeri, "Prediction of bubble lift-off diameter in subcooled flow boiling using AVL fire," *Elixir Thermal Engineering,* vol. 56, pp. 13715-13718, 2013.

[29] V. Prodanovic, D. Fraser and M. Salcudean, "Bubble Behavoir in Subcooled Flow Boiling of Water at Low Pressures and Low Flow Rates," *International Journal of Multiphase Flow,* vol. 28, pp. 1-19, 2002.

[30] G. Sateesh, S. K. Das and A. R. Balakrishnan, "Analysis of pool boiling heat transfer: effect of bubbles sliding on the heating surface," *International Journal of Heat and Mass Transfer,* vol. 48, pp. 1543-1553, 2005.

[31] R. Situ, Y. Mi, M. Ishii and M. Mori, "Photographic study of bubble behaviors in forced convection subcooled boiling," *International Journal of Heat and Mass Transfer,* vol. 47, pp. 3659-3667, 2004.

[32] V. H. DelValle and D. Kenning, "Subcooled Flow Boiling in High Heat Flux," *International Journal of Heat and Mass Transfer,* vol. 28, no. 10, pp. 1907-1920, 1985.

[33] H. Kim and J. Buongiorno, "A Novel Infrared-Based Experimental Technique to Detect Phase Dynamics on Boiling Surfaces," in *Proceedings of the 11th International Topical Meeting on Nuclear Reactor Thermal-Hydraulics (NURETH-14)*, 2011.

[34] C. Gerardi, J. Buongiorno, L. Hu and T. McKrell, "Study of Bubble Growth in Water Pool Boiling Through Synchronized, Infrared Thermometry and High-Speed Video," *International Journal of Heat and Mass Transfer,* vol. 98, no. 4, pp. 4185-4192, 2010.

[35] X. Duan, B. Phillips, T. McKrell and J. Buongiorno, "Synchronized high-speed video, infrared thermometry, and particle image velocimetry data for validation of interface-tracking simulations of nucleate boiling phenomena," *Experimental Heat Transfer,* vol. 26, pp. 169-197, 2013.

[36] M. G. Cooper and A. P. Lloyd, "The microlayer in nucleate pool boiling," *International Journal of Heat and Mass Transfer,* vol. 12, no. 8, pp. 895-913, 1969.

[37] T. Cebeci and P. Bradshaw, Momentum Transfer in Boundary Layers, Hemisphere/McGraw-Hill, 1977.

[38] H. Schlichting and K. Gersten, Boundary Layer Theory, New York: Springer, 2000.

[39] D. Chatzikyriakou, D. Caviezel, J. Buongiorno and D. Lakehal, "ITM Test Case 1: LES and DNS of turbulent flow in a closed channel with flat walls featuring a pattern of small hemispherical solid obstacles," Cambridge, MA, 2011.

[40] J. Kim, "Review of Nucleate Pool Boiling Heat Transfer Mechanisms," *International Journal of Multiphase Flow,* vol. 35, no. 12, pp. 1067-1076, 2009.

[41] J. F. Klausner, R. Mei, D. Bernhard and L. Zeng, "Vapor Bubble Departure in Forced Convection Boiling," *International Journal of Heat and Mass Transfer,* vol. 36, no. 3, pp. 651-662, 1993.

[42] B. Yun, A. Splawski, S. Lo and C.-H. Song, "Advanced Wall Boiling Model for the Subcooled Boiling Flow with CFD Code," in *The Seventh Korea-Japan Symposium on Nuclear Thermal Hydraulics and Safety*, Chuncheon, Korea, 2010.

[43] R. Sugrue and J. Buongiorno, "A Modified Force-Balance Model for Predicting Bubble Departure Diameter in Subcooled Flow Boiling," in *NURETH-15*, Pisa, Italy, May 12-15, 2013.

[44] S. Maity, Effect of Velocity and Gravity on Bubble Dynamics, University of California, Los Angeles: MS Thesis, 2000.

[45] C. B. Jenssen, T. Kvamdal, H. Andersson, B. Pettersen, P. Fox, A. Ecer, N. Satofuka and J. Periaux, Parallel Computational Fluid Dynamics 2000: Trends and Applications, Elsevier, 2001.

[46] G. Tryggvason, R. Scardovelli and S. Zaleski, Direct Numerical SSimulation of Gas-Liquid Multiphase Flows, Cambridge University Press, 2011.

[47] N. Zuber, "The Dynamics of Vapor Bubbles in nonuniform temperature fields," *International Journal of Heat and Mass Transfer,* vol. 2, pp. 83-98, 1961.

7 Appendix: Implementation in STAR-CCM+

The models described are implemented in STAR-CCM+ by linking user libraries compiled from subroutines written in C on a linux OS. These subroutines for the bubble departure diameter and bubble departure frequency are attached to this report. There are four files for each subroutine: uclib.h, uclib.c, ucmodels.h, and ucmodels.h.

7.1 Bubble Departure Diameter

```
/* ---------------------------------------------------------------------------
 * uclib.h: registration for any ucode models
 * ---------------------------------------------------------------------------
 *
 *
 * Library Build Command (linux)
 * ---------------------------------------------
 *
 *   gcc -fPIC -shared *.c -o libuser.so
 *
 *
 * Function Name Registration
 * ---------------------------------------------
 *
 *   ucfunc(
 *     void *func,
 *     char *type ,
 *     char *name
 *   );
 *
 *   function type can be
 *
 *     "BoundaryProfile"
 *     "RegionProfile"
 *     "ScalarFieldFunction"
 *     "VectorFieldFunction"
 *
 *
 * Function Argument Registration
 * ---------------------------------------------
 *
 *   ucarg (
 *     void *func,
 *     char *type,
 *     char *variable,
 *     int   size
 *   );
 *
 *   argument type can be
 *
 *     "Cell"
```

```
*    "Face"
*
*   variable names are scoped names of the form
*   below rather than usual user names
*
*    "Pressure"
*    "Phase0::VolumeFraction"
*    "Phase0::Density"
*    "Phase0::U_Velocity"
*    "PhaseInteraction0::InteractionLengthScale"
*
*   also note that, even though Velocity is returned from
*   user functions as a vector, for input arguments
*   Velocity is passed as individual components,
*   with variable names
*
*     "U_Velocity"
*     "V_Velocity"
*     "W_Velocity"
*
*   finally note that not all fields may be available
*   during initialisation
*
*   size is one of
*
*     sizeof(Real)
*     sizeof(CoordReal)   --> Field functions are of this type
*     sizeof(PressureReal)
*
*
* C-function arguments
* ----------------------------------------------
* Elemental Type            Declaration
* ----------------------------------------------
*
* int                       int *arg
*
* PressureReal              PressureReal *arg
*
* Real                      Real *arg
*
* unsigned int              int *arg
*
* Vector<2, unsigned int>   int (*arg)[2]
*
* Vector<3, CoordReal>      CoordReal (*arg)[3]
*
* Vector<3, Real>           Real (*arg)[3]
```

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

#ifndef UCLIB_H
#define UCLIB_H

#define UCFUNC_TYPE_SCALAR_FF "ScalarFieldFunction"
#define UCFUNC_TYPE_VECTOR_FF "VectorFieldFunction"
#define UCFUNC_TYPE_SCALAR_RP "RegionProfile"
#define UCFUNC_TYPE_VECTOR_RP "RegionProfile"
#define UCFUNC_TYPE_SCALAR_BP "BoundaryProfile"

typedef float  Real;
typedef double CoordReal;
typedef double PressureReal;

#ifdef __cplusplus
extern "C" {
#endif

#if defined(WIN32) || defined(_WINDOWS) || defined(_WINNT)
# define USERFUNCTION_EXPORT __declspec(dllexport)
# define USERFUNCTION_IMPORT __declspec(dllimport)
#else
# define USERFUNCTION_EXPORT
# define USERFUNCTION_IMPORT
#endif

extern void USERFUNCTION_IMPORT ucarg(void *, char *, char *, int);
extern void USERFUNCTION_IMPORT ucfunc(void *, char *, char *);
extern void USERFUNCTION_IMPORT ucfunction(void *, char *, char *, int, ...);

void USERFUNCTION_EXPORT uclib();

#ifdef __cplusplus
}
#endif

#endif
```

```
/* ---------------------------------------------------------------------------
 * uclib.c: registration for shareable application-specific ucode models
 * ---------------------------------------------------------------------------
 */

#include "ucfbmodels.h"

void uclib()
{

/*****************************************************************************
**
 * userBubbleDiameter

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

  ucfunc(
    /* void *func     */ userBubbleDiameter,
    /* char *type     */ UCFUNC_TYPE_SCALAR_BP,
    /* char *name     */ UCFUNC_TAG_BUBBLE_DIAMETER
  );

ucarg (
    /* void *func     */ userBubbleDiameter,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_SURFACE_TENSION,
    /* int   size     */ sizeof(CoordReal)
  );

  ucarg (
    /* void *func     */ userBubbleDiameter,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_LIQUID_DENSITY,
    /* int   size     */ sizeof(Real)
  );

ucarg (
    /* void *func     */ userBubbleDiameter,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_VAPOR_DENSITY,
    /* int   size     */ sizeof(Real)
  );

ucarg (
    /* void *func     */ userBubbleDiameter,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_DYNAMIC_VISCOSITY,
```

```
    /* int   size     */ sizeof(CoordReal)
  );

ucarg (
    /* void *func     */ userBubbleDiameter,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_U_TAU,
    /* int   size     */ sizeof(CoordReal)
  );

//ucarg (
    /* void *func     */ //userBubbleDiameter,
    /* char *type     */ //"Cell",
    /* char *variable */ //UCARG_TAG_PRANDTL,
    /* int   size     */ //sizeof(CoordReal)
 // );

ucarg (
    /* void *func     */ userBubbleDiameter,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_ADVANCING_CONTACT_ANGLE,
    /* int   size     */ sizeof(CoordReal)
 );

ucarg (
    /* void *func     */ userBubbleDiameter,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_SPECIFIC_HEAT,
    /* int   size     */ sizeof(CoordReal)
  );


ucarg (
    /* void *func     */ userBubbleDiameter,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_HEAT_FLUX,
    /* int   size     */ sizeof(CoordReal)
  );

ucarg (
    /* void *func     */ userBubbleDiameter,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_RECEDING_CONTACT_ANGLE,
    /* int   size     */ sizeof(CoordReal)
  );

ucarg (
    /* void *func     */ userBubbleDiameter,
```

```
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_THETA,
    /* int   size     */ sizeof(CoordReal)
  );

ucarg (
    /* void *func     */ userBubbleDiameter,
    /* char *type     */ "Cell",
```

```
/* ---------------------------------------------------------------------------
 * ucmodels.h: implementation of shareable application-specific ucode models
 * ---------------------------------------------------------------------------
 */

#ifndef UCFBMODELS_H
#define UCFBMODELS_H

#include "uclib.h"

/* output field presentation names */

#define UCFUNC_TAG_BUBBLE_DIAMETER "FB_Bubble_Diameter_Model"


/* input field solver names */

// This field function looks strange in STAR:
#define UCARG_TAG_SURFACE_TENSION "$SurfaceTensionPhaseInteraction"
#define UCARG_TAG_LIQUID_DENSITY "Phase0::Density"
#define UCARG_TAG_VAPOR_DENSITY "Phase1::Density"
#define UCARG_TAG_DYNAMIC_VISCOSITY "$DynamicViscosityLiquid"
#define UCARG_TAG_U_TAU "$UstarLiquid"
#define UCARG_TAG_ADVANCING_CONTACT_ANGLE "$Alpha"
#define UCARG_TAG_SPECIFIC_HEAT "$SpecificHeatLiquid"
#define UCARG_TAG_THERMAL_DIFFUSIVITY "$ThermalDiffusivity"
#define UCARG_TAG_HEAT_FLUX "$BoundaryHeatFluxLiquid"
#define UCARG_TAG_RECEDING_CONTACT_ANGLE "$Beta"
#define UCARG_TAG_THETA "$theta"
#define UCARG_TAG_THERMAL_CONDUCTIVITY "$ThermalConductivityLiquid"
#define UCARG_TAG_TAU_WALL_LIQUID "$Tau_wall"
#define UCARG_TAG_LIQUID_TEMPERATURE "$TemperatureLiquid"
#define UCARG_TAG_SATURATION_TEMPERATURE "$TemperaturePhaseInteraction"
#define UCARG_TAG_LATENT_HEAT "$LatentHeatPhaseInteraction"


/* function prototypes */

void userBubbleDiameter(
  Real*        result,
  int          size,
  CoordReal*   SurfaceTensionPhaseInteraction,
  Real*        DensityLiquid,
  Real*        DensityVapor,
  CoordReal*   DynamicViscosityLiquid,
  CoordReal*   UstarLiquid,
  CoordReal*   Alpha,
  CoordReal*   SpecificHeatLiquid,
```

```
 // CoordReal* ThermalDiffusivity,
  CoordReal*   BoundaryHeatFluxLiquid,
  CoordReal*   Beta,
  CoordReal*   theta,
  CoordReal*   ThermalConductivityLiquid,
 // CoordReal* Tau_wall,
  CoordReal*   TemperatureLiquid,
  CoordReal*   TemperaturePhaseInteraction,
  CoordReal*   LatentHeatPhaseInteraction

) ;


#endif
```

```
/* ---------------------------------------------------------------------------
 * ucmodels.c: implementation for shareable application-specific ucode models
 * ---------------------------------------------------------------------------
 */

#include <math.h>
#include <stdio.h>
#include "ucfbmodels.h"

/*****************************************************************************
**
 * userBubbleDiameter(bubbleVolumeFraction).

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

void userBubbleDiameter(
  Real*        result,
  int          size,
  CoordReal*   SurfaceTensionPhaseInteraction,
  Real*        DensityLiquid,
  Real*        DensityVapor,
  CoordReal*   DynamicViscosityLiquid,
  CoordReal*   UstarLiquid,
  CoordReal*   Alpha,
  CoordReal*   SpecificHeatLiquid,
 // CoordReal* ThermalDiffusivity,
  CoordReal*   BoundaryHeatFluxLiquid,
  CoordReal*   Beta,
  CoordReal*   theta,
  CoordReal*   ThermalConductivityLiquid,
  //CoordReal* Tau_wall,
  CoordReal*   TemperatureLiquid,
  CoordReal*   TemperaturePhaseInteraction,
  CoordReal*   LatentHeatPhaseInteraction

) {

  /* [0] declarations */
// Defining Constants in the Bubble Departure Model //

// Surface Tension Force //

double pi        =        3.1416;                    // Defining the constant pi; This may already be
defined in C by using
//double alpha   = 90.63*(pi/180);  // Advancing contact angle in Rosie's Experiment [rad] //
```

```
//double beta =          8.03*(pi/180); // Receding contact angle in Rosie's Experiment [rad] //

// Buoyancy Force //

double g =             9.81; // Gravitational Constant [m/s^2] //
double v_l =           0.4174;          // Bulk Fluid Velocity [m/s] //


// Quasi-Steady Drag Force //

double Kappa = 0.4;   // Constant provided by Klausner //
double Chi =   11;       // Constant Provided by Klausner //
double c =             7.4;  // Constant Provided by Klausner //
double n =             0.65; // Constant Provided by Klausner //

// Unsteady Force //

double b = 1.56;         // Constant provided by Zuber //
int        s = 2;          // Constant Provided by Yun //
double phi = pi/18; // Inclination angle of the growing bubble; Provided by Klausner //
double C_s = 1;                  // Variable constant in the model; Yun uses C_s=1 //

  /* [1] record user model activation and run parameters */

 /* Calculation of Forces following Rosie's Implemenation
NOTE: Unsteady Force
Yun's modification to include the effect of subcooling on the top of the bubble is not included
used */

// NOTE: R is the current guess for the bubble radius. This will loop through a pre-determined
set of lengths. //

// Place constants before the radius and time loops //

double delta_R = 5E-7; // The change in the guess radius R for the loop [m]
double R; // Bubble radius for departure [m]

double t_max = 1; // The maximum time for the time loop [s]



// First Loop through all the cells
int i;
for (i = 0; i != size; ++i)
        {

        double Theta = theta[i];
        double alpha = Alpha[i]*(pi/180);  // in radians
```

```
        double beta = Beta[i]*(pi/180);                    // in radians
        double sigma = SurfaceTensionPhaseInteraction[i];
        double mu_l = DynamicViscosityLiquid[i];
        double nu_l = (mu_l)/(DensityLiquid[i]);
        double c_pl = SpecificHeatLiquid[i];
        double h_fg = LatentHeatPhaseInteraction[i];
        double k_l = ThermalConductivityLiquid[i];
        float rho_l = DensityLiquid[i];
        float rho_v = DensityVapor[i];
        double PR = c_pl*mu_l/k_l;
        double eta_l = k_l/(rho_l*c_pl);

// Checking the forces when R = 1E-5
for (R = 1E-5; R <= 1E-1; R = R + delta_R)
{

// Then the calculation for u_star:
        double u_star = UstarLiquid[i];
        double y_R = R*u_star/nu_l; // The non-dimensionalized position at y = R //

        // Calculate U
        double U = u_star/Kappa*log(1 + Kappa*y_R) + c*u_star*(1-exp(-y_R/Chi)-
y_R/Chi*exp(-0.33*y_R));


        // Also calculate dU_dY manually:
        double dU_dY =
(u_star*u_star/(nu_l*(1+Kappa*R*u_star/nu_l))+c*(u_star*u_star*exp(-
R*u_star/nu_l/Chi)/nu_l/Chi-u_star*u_star*exp(-
0.33*R*u_star/nu_l)/nu_l/Chi+0.33*R*u_star*u_star*u_star*exp(-
0.33*R*u_star/nu_l)/(nu_l*nu_l)/Chi));

/* ------------------------------------------------------------
------------------Surface Tension Force---------------------------
-----------------------------------------------------------------*/

// NOTE: R is the current guess for the bubble radius. This will loop through a pre-determined
set of lengths. //

int i_dw = 40;          // The current number for the ratio of the foot diameter to the bubble
diameter so that: "d_w/d_b = 1/i_dw" //
double F_sx, F_sy;     // The x and y-direction surface tension forces respectively

double d_w = 2*R/i_dw; // Calculation for the bubble foot diameter //

F_sx = (-1.25*d_w*sigma*pi*(alpha-beta)/(pi*pi-(alpha-beta)*(alpha-
beta)))*(sin(alpha)+sin(beta));
F_sy = -d_w*sigma*pi*(cos(beta) - cos(alpha))/(alpha-beta);
```

```
/* ---------------------------------------------------------------
------------------Buoyancy Force ---------------------------------
-----------------------------------------------------------------*/


double F_b;

F_b = 4*pi*(R*R*R)*(rho_l - rho_v)*g/3;

/* ---------------------------------------------------------------
------------------Contact Pressure Force -------------------------
-----------------------------------------------------------------*/

double r_r = 5*R; // Radius of curvature of the bubble at the reference point on the surface y=0 //

double F_cp;

F_cp = pi*(d_w*d_w)*2*sigma/(4*r_r);


/* ---------------------------------------------------------------
------------------Quasi-Steady Drag Force -------------------------
-----------------------------------------------------------------*/

double Re_b; // Reynolds number for the bubble and Uniform flow velocity taken at the bubble
diameter //

Re_b = 2*U*R/nu_l;

double F_qs;
double inner_pow1 = pow((12/Re_b),n);
double inner_pow2 = pow(0.796,n);
double inner_pow3 = pow((inner_pow1+inner_pow2),(-1/n));

F_qs = 6*pi*nu_l*rho_l*U*R*(0.666667+inner_pow3);

/* ---------------------------------------------------------------
------------------Unsteady Force ---------------------------------
-----------------------------------------------------------------*/

// NOTE: This is using Yun's model //

double Ja;
double T_sat = TemperaturePhaseInteraction[i];
double T_wall = TemperatureLiquid[i];

double DeltaT_sat = T_wall-T_sat;
```

```
if (DeltaT_sat <= 0)
        { result[i] = 0;
        break;
        }

double F_du = 0;
double F_dux = 0;
double F_duy = 0;

Ja = rho_l*c_pl*(DeltaT_sat)/(rho_v*h_fg);

/* -------------------------------------
Include this for Yun's subcooling:
Nu = 2 + 0.6*Re_b^0.5*PR^0.3;
h_i = Nu*k_l/(2*R);
q_l = h_i*(T_sat - T_liq);
------------------------------------- */

double R_t; // Bubble growth rate, first derivative, second derivative
double R_tt;
double R_ttt;

double t;
// Solve for time explicitly instead of using a loop:
 t = R*R*pi/(4*b*b*Ja*Ja*eta_l);

R_t = 2*b*Ja*sqrt(eta_l*t/pi);
R_tt  = b*Ja*sqrt(eta_l/(pi*t)); // include this for Yun's subcooling: - b*q_l/(s*h_fg*rho_v);
R_ttt = -b*Ja*sqrt(eta_l/pi)*(pow(t,(-1.5)))/2;


F_du = -rho_l*pi*pow(R_t,2)*(R_t*R_ttt + 1.5*C_s*pow(R_tt,2));

F_dux = sin(phi)*F_du;
F_duy = cos(phi)*F_du;

/* ---------------------------------------------------------------
------------------Hydrodynamic Force ----------------------------
-----------------------------------------------------------------*/

double F_h;

F_h = 9*rho_l*U*U*pi*(d_w*d_w)/32;

/* ---------------------------------------------------------------
------------------Shear Lift Force -------------------------------
-----------------------------------------------------------------*/
```

```
double G_s; // Dimensionless shear rate of oncoming flow

G_s = dU_dY*R/U;

double F_sL;

double part_1 = 0.5*rho_l*U*U*pi*(R*R);
double inpow = (1/(Re_b*Re_b) + 0.014*G_s*G_s);

F_sL = part_1*3.877*pow(G_s,0.5)*pow(inpow,0.25);

//fprintf( stderr, "ucfbmodels.c: userBubbleDiameter: F_sL=%e\n", F_sL );

/* ---------------------------------------------------------------
------------------Summation of Forces ----------------------------
-------------------------------------------------------------------*/

double Sum_Fx, Sum_Fy;

Sum_Fx = F_sx  + F_qs + F_b*sin(Theta) + F_dux;
Sum_Fy = F_sy + F_sL - F_b*cos(Theta) - F_h + F_duy + F_cp;

// If the sum of y-forces is ~zero  ==> the bubble departs!
       if (Sum_Fx > 1E-8)       // approx. 1e-8=0
          {                    result[i] = 2*R;
                    break;
                    }


// If sum of x-forces is ~zero  ==> the bubble slides!
      if(Sum_Fy> 1E-8)     // approx. 1e-8=0
        {           result[i] = 2*R;
                    break;
                    }


}
//the break by the addition of forces gets you out of the Radius-loop and then you go to the next
cell (i)
}

}
```

7.2 Bubble Departure Frequency

```
/* ---------------------------------------------------------------------------
 * uclib.h: registration for any ucode models
 * ---------------------------------------------------------------------------
 *
 *
 * Library Build Command (linux)
 * ---------------------------------------------
 *
 *   gcc -fPIC -shared *.c -o libuser.so
 *
 *
 * Function Name Registration
 * ---------------------------------------------
 *
 *   ucfunc(
 *     void *func,
 *     char *type ,
 *     char *name
 *   );
 *
 *   function type can be
 *
 *     "BoundaryProfile"
 *     "RegionProfile"
 *     "ScalarFieldFunction"
 *     "VectorFieldFunction"
 *
 *
 * Function Argument Registration
 * ---------------------------------------------
 *
 *   ucarg (
 *     void *func,
 *     char *type,
 *     char *variable,
 *     int   size
 *   );
 *
 *   argument type can be
 *
 *     "Cell"
 *     "Face"
 *
 *   variable names are scoped names of the form
 *   below rather than usual user names
 *
 *    "Pressure"
```

```
*    "Phase0::VolumeFraction"
*    "Phase0::Density"
*    "Phase0::U_Velocity"
*    "PhaseInteraction0::InteractionLengthScale"
*
*   also note that, even though Velocity is returned from
*   user functions as a vector, for input arguments
*   Velocity is passed as individual components,
*   with variable names
*
*     "U_Velocity"
*     "V_Velocity"
*     "W_Velocity"
*
*   finally note that not all fields may be available
*   during initialisation
*
*   size is one of
*
*     sizeof(Real)
*     sizeof(CoordReal)   --> Field functions are of this type
*     sizeof(PressureReal)
*
*
* C-function arguments
* ----------------------------------------------
* Elemental Type            Declaration
* ----------------------------------------------
*
* int                       int *arg
*
* PressureReal              PressureReal *arg
*
* Real                      Real *arg
*
* unsigned int              int *arg
*
* Vector<2, unsigned int>   int (*arg)[2]
*
* Vector<3, CoordReal>      CoordReal (*arg)[3]
*
* Vector<3, Real>           Real (*arg)[3]
*
* ----------------------------------------------
*
*/

#ifndef UCLIB_H
```

```
#define UCLIB_H

#define UCFUNC_TYPE_SCALAR_FF "ScalarFieldFunction"
#define UCFUNC_TYPE_VECTOR_FF "VectorFieldFunction"
#define UCFUNC_TYPE_SCALAR_RP "RegionProfile"
#define UCFUNC_TYPE_VECTOR_RP "RegionProfile"
#define UCFUNC_TYPE_SCALAR_BP "BoundaryProfile"

typedef float  Real;
typedef double CoordReal;
typedef double PressureReal;

#ifdef __cplusplus
extern "C" {
#endif

#if defined(WIN32) || defined(_WINDOWS) || defined(_WINNT)
# define USERFUNCTION_EXPORT __declspec(dllexport)
# define USERFUNCTION_IMPORT __declspec(dllimport)
#else
# define USERFUNCTION_EXPORT
# define USERFUNCTION_IMPORT
#endif

extern void USERFUNCTION_IMPORT ucarg(void *, char *, char *, int);
extern void USERFUNCTION_IMPORT ucfunc(void *, char *, char *);
extern void USERFUNCTION_IMPORT ucfunction(void *, char *, char *, int, ...);

void USERFUNCTION_EXPORT uclib();

#ifdef __cplusplus
}
#endif

#endif
```

```
/* ---------------------------------------------------------------------------
 * uclib.c: registration for shareable application-specific ucode models
 * ---------------------------------------------------------------------------
 */

#include "ucmodels.h"

void uclib()
{

/*****************************************************************************
**
 * userBubbleDepartureFrequency

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

  ucfunc(
    /* void *func     */ userFrequency,
    /* char *type     */ UCFUNC_TYPE_SCALAR_BP,
    /* char *name     */ UCFUNC_TAG_BUBBLE_DEPARTURE_FREQUENCY
  );

  ucarg (
    /* void *func     */ userFrequency,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_LIQUID_DENSITY,
    /* int   size     */ sizeof(Real)
  );

ucarg (
    /* void *func     */ userFrequency,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_VAPOR_DENSITY,
    /* int   size     */ sizeof(Real)
  );

ucarg (
    /* void *func     */ userFrequency,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_NUCLEATION_SITES,
    /* int   size     */ sizeof(CoordReal)
  );

ucarg (
    /* void *func     */ userFrequency,
    /* char *type     */ "Face",
```

```
    /* char *variable */ UCARG_TAG_BUBBLE_DIAMETER,
    /* int   size     */ sizeof(CoordReal)
  );

ucarg (
    /* void *func     */ userFrequency,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_LIQUID_TEMPERATURE_CELL,
    /* int   size     */ sizeof(CoordReal)
  );


ucarg (
    /* void *func     */ userFrequency,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_LATENT_HEAT,
    /* int   size     */ sizeof(CoordReal)
  );

ucarg (
    /* void *func     */ userFrequency,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_DYNAMIC_VISCOSITY,
    /* int   size     */ sizeof(CoordReal)
  );

ucarg (
    /* void *func     */ userFrequency,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_U_TAU,
    /* int   size     */ sizeof(CoordReal)
  );

ucarg (
    /* void *func     */ userFrequency,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_SATURATION_TEMPERATURE,
    /* int   size     */ sizeof(CoordReal)
  );

ucarg (
    /* void *func     */ userFrequency,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_LIQUID_TEMPERATURE_WALL,
    /* int   size     */ sizeof(CoordReal)
  );

ucarg (
```

```
    /* void *func     */ userFrequency,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_SPECIFIC_HEAT,
    /* int   size     */ sizeof(CoordReal)
  );

ucarg (
    /* void *func     */ userFrequency,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_THERMAL_CONDUCTIVITY,
    /* int   size     */ sizeof(CoordReal)
  );

ucarg (
    /* void *func     */ userFrequency,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_SURFACE_TENSION,
    /* int   size     */ sizeof(CoordReal)
  );

ucarg (
    /* void *func     */ userFrequency,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_ADVANCING_CONTACT_ANGLE,
    /* int   size     */ sizeof(CoordReal)
 );

ucarg (
    /* void *func     */ userFrequency,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_RECEDING_CONTACT_ANGLE,
    /* int   size     */ sizeof(CoordReal)
  );



}
```

```
/* ---------------------------------------------------------------------------
 * ucmodels.h: implementation of shareable application-specific ucode models
 * ---------------------------------------------------------------------------
 */

#ifndef UCMODELS_H
#define UCMODELS_H

#include "uclib.h"

/* output field presentation names */

#define UCFUNC_TAG_BUBBLE_DEPARTURE_FREQUENCY "Frequency"


/* input field solver names */

// This field function looks strange in STAR:
#define UCARG_TAG_LIQUID_DENSITY "Phase0::Density"
#define UCARG_TAG_VAPOR_DENSITY "Phase1::Density"
#define UCARG_TAG_NUCLEATION_SITES
"$NucleationSiteNumberDensityPhaseInteraction"
#define UCARG_TAG_BUBBLE_DIAMETER "$BubbleDepartureDiameterPhaseInteraction"
#define UCARG_TAG_LIQUID_TEMPERATURE_CELL "$TemperatureLiquid"
#define UCARG_TAG_LATENT_HEAT "$LatentHeatPhaseInteraction"
#define UCARG_TAG_DYNAMIC_VISCOSITY "$DynamicViscosityLiquid"
#define UCARG_TAG_U_TAU "$UstarLiquid"
#define UCARG_TAG_SATURATION_TEMPERATURE "$TemperaturePhaseInteraction"
#define UCARG_TAG_LIQUID_TEMPERATURE_WALL "$TemperatureLiquid"
#define UCARG_TAG_SPECIFIC_HEAT "$SpecificHeatLiquid"
#define UCARG_TAG_THERMAL_CONDUCTIVITY "$ThermalConductivityLiquid"
#define UCARG_TAG_SURFACE_TENSION "$SurfaceTensionPhaseInteraction"
#define UCARG_TAG_ADVANCING_CONTACT_ANGLE "$Alpha"
#define UCARG_TAG_RECEDING_CONTACT_ANGLE "$Beta"




/* function prototypes */

void userFrequency(
  Real*        result,
  int          size,
  Real*            DensityLiquid,
  Real*            DensityVapor,
  CoordReal*   NucleationSiteNumberDensityPhaseInteraction,
  CoordReal*   BubbleDepartureDiameterPhaseInteraction,
```

```
  CoordReal*   TemperatureLiquidCell,
  CoordReal*   LatentHeatPhaseInteraction,
  CoordReal*   DynamicViscosityLiquid,
  CoordReal*   UstarLiquid,
  CoordReal*   TemperaturePhaseInteraction,
  CoordReal*   TemperatureLiquidWall,
  CoordReal*   SpecificHeatLiquid,
  CoordReal*   ThermalConductivityLiquid,
  CoordReal*   SurfaceTensionPhaseInteraction,
  CoordReal*   Alpha,
  CoordReal*   Beta


) ;


#endif
```

```
/* ---------------------------------------------------------------------------
 * ucmodels.c: implementation for shareable application-specific ucode models
 * ---------------------------------------------------------------------------
 */

#include <math.h>
#include <stdio.h>
#include "ucmodels.h"

/*****************************************************************************
**
 * userBubbleDepartureFrequency

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

void userFrequency(
  Real*        result,
  int          size,
  Real*            DensityLiquid,
  Real*            DensityVapor,
  CoordReal*   NucleationSiteNumberDensityPhaseInteraction,
  CoordReal*   BubbleDepartureDiameterPhaseInteraction,
  CoordReal*   TemperatureLiquidCell,
  CoordReal*   LatentHeatPhaseInteraction,
  CoordReal*   DynamicViscosityLiquid,
  CoordReal*   UstarLiquid,
  CoordReal*   TemperaturePhaseInteraction,
  CoordReal*   TemperatureLiquidWall,
  CoordReal*   SpecificHeatLiquid,
  CoordReal*   ThermalConductivityLiquid,
  CoordReal*   SurfaceTensionPhaseInteraction,
  CoordReal*   Alpha,
  CoordReal*   Beta

) {

  /* [0] declarations */
// Defining Constants in the Bubble Departure Model //

// Surface Tension Force //

double pi        =        3.1416;                    // Defining the constant pi; This may already be
defined in C by using

// Constants for model //

double b     = 1.56;      // Constant provided by Zuber //
```

```
int    s     = 2;    // Constant Provided by Yun //
double phi   = pi/18;    // Inclination angle of the growing bubble; Provided by Klausner //
double C_s   = 1;          // Variable constant in the model; Yun uses C_s=1 //

double Kappa = 0.4;       // Constant provided by Klausner //
double Chi =   11;         // Constant Provided by Klausner //
double c =     7.4;       // Constant Provided by Klausner //
double n =     0.65;      // Constant Provided by Klausner //

  /* [1] record user model activation and run parameters */


// First Loop through all the cells
int i;
for (i = 0; i != size; ++i)
          {

// Call field functions from simulation

          double N = NucleationSiteNumberDensityPhaseInteraction[i];
          double D_b = BubbleDepartureDiameterPhaseInteraction[i];
          double h_fg = LatentHeatPhaseInteraction[i];
          float rho_l = DensityLiquid[i];
          float rho_v = DensityVapor[i];

          double mu_l = DynamicViscosityLiquid[i];
          double u_star = UstarLiquid[i];
          double T_sat = TemperaturePhaseInteraction[i];
          double T_wall = TemperatureLiquidWall[i];
          double T_liq = TemperatureLiquidCell[i];
          double c_pl = SpecificHeatLiquid[i];
          double k_l = ThermalConductivityLiquid[i];

          double alpha = Alpha[i]*(pi/180);  // in radians
          double beta = Beta[i]*(pi/180);                    // in radians
          double sigma = SurfaceTensionPhaseInteraction[i];

          double R = D_b/2;
          double Delta_max = 2E-6; // the maximum height of the microlayer is 2 microns

/* ---------------------------------------------------------------
------------------Growth time ----------------------------------
-----------------------------------------------------------------*/

double nu_l = (mu_l)/(rho_l);
double PR = c_pl*mu_l/k_l;
double eta_l = k_l/(rho_l*c_pl);
```

```
double y_R = R*u_star/nu_l; // The non-dimensionalized position at y = R //
double U = u_star/Kappa*log(1 + Kappa*y_R) + c*u_star*(1-exp(-y_R/Chi)-y_R/Chi*exp(-
0.33*y_R));

double Re_b; // Reynolds number for the bubble and Uniform flow velocity taken at the bubble
diameter //
Re_b = 2*U*R/nu_l;


double Ja;
double DeltaT_sat = T_wall-T_sat;

Ja = rho_l*c_pl*(DeltaT_sat)/(rho_v*h_fg);

// Include this for Yun's subcooling:
double Nu = 2 + 0.6*sqrt(Re_b)*pow(PR,0.3);
double h_i = Nu*k_l/(2*R);
double q_l = h_i*(T_sat - T_liq);

// Solve for t since we now R:
double t_1 = 0;
double t_2 = 0;
double inner_1 = pow(b*h_fg*Ja*rho_v*s,4)*eta_l*eta_l;
double inner_2 = pow(b*h_fg*rho_v*s,3)*Ja*Ja*eta_l*pi*q_l*R;


// for including the subcooing term:
/*if (q_l > 0 && (inner_1-inner_2) > 0)
        {t_1 = ((2*b*b*h_fg*h_fg*Ja*Ja*eta_l*rho_v*rho_v*s*s) - /* this can be a plus or
minus */ /* 2*sqrt(inner_1-inner_2) - b*h_fg*pi*q_l*rho_v*R*s)/(b*b*pi*q_l*q_l);} */

// solve for t without using the subcooling portion of the bubble growth model: This is also what
Yeoh and Tu (2005) did.
t_2 = R*R*pi/(4*b*b*Ja*Ja*eta_l);


/* ---------------------------------------------------------------
------------------Wait time----------- -------------------------
-----------------------------------------------------------------*/

// Calculated using Yeoh and Tu (2011) Fractal Model Method

// Calculate the critical radius of the cavity

double D_c, R_c;
D_c = 4*sigma*T_sat/(rho_v*h_fg*DeltaT_sat);

R_c = D_c/2;
```

```
// Calculate variables in model

C_1 = (1+cos(beta))/sin(beta);
C_2 = 1/sin(beta);

// Calculate the wait time:
double t_w, inner_3;

inner_3 = ((T_wall-T_liq)*R_c*C_1)/(DeltaT_sat - 2*sigma*T_sat/(C_2*R_c*rho_v*h_fg));

t_w = inner_3*inner_3/(eta_l*pi);

/* ---------------------------------------------------------------
------------------Summation of Wait and Growth Times---------------
-------------------------------------------------------------------*/
double f;

f = 1/(t_w+t_2);

result[i] = f;



        }

}
```