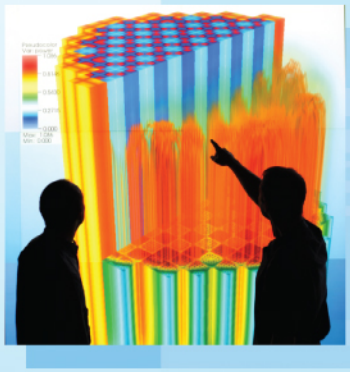
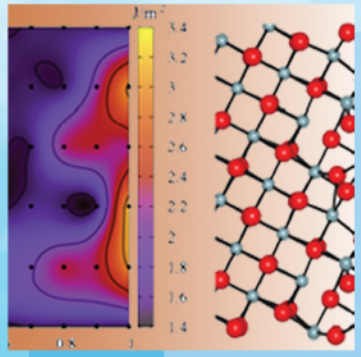
CASL-U-2014-0205-000

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.P9.08

Development of a General Purpose Subgrid Wall Boiling Model from Improved Physical Understanding for Use in Computational Fluid Dynamics

Emilio Baglietto
Massachusetts Institute of Technology

September 30, 2014

CASL-U-2014-0205-000

U.S. DEPARTMENT OF ENERGY | Nuclear Energy

Development of a General Purpose Subgrid Wall Boiling Model from Improved Physical Understanding for Use in Computational Fluid Dynamics

by
Lindsey Anne Gilman

B.S. Chemistry, Valparaiso University, 2010
S.M. Nuclear Science and Engineering, Massachusetts Institute of Technology, 2012

SUBMITTED TO THE DEPARTMENT OF NUCLEAR SCIENCE AND ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY IN NUCLEAR SCIENCE AND ENGINEERING
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2014

© 2014 Massachusetts Institute of Technology
All rights reserved

Author: _____

Lindsey Gilman
Department of Nuclear Science and Engineering
May 14, 2014

Certified by: _____

Emilio Baglietto, Ph.D.
Assistant Professor of Nuclear Science and Engineering
Thesis Supervisor

Certified by: _____

Jacopo Buongiorno, Ph.D.
Associate Professor of Nuclear Science and Engineering
Thesis Reader

Certified by: _____

Evelyn N. Wang, Ph.D.
Associate Professor of Mechanical Engineering
Thesis Reader

Certified by: _____

Paolo Di Marco, Ph.D.
Associate Professor, Engineering Thermodynamics and Heat Transfer, School of Engineering
University of Pisa
Thesis Reader

Accepted by: _____

Mujid Kazimi
TEPCO Professor of Nuclear Engineering
Chair, Department Committee on Graduate Students

CASL-U-2014-0205-000

Development of a General Purpose Subgrid Wall Boiling Model from Improved Physical Understanding for Use in Computational Fluid Dynamics

by
Lindsey Anne Gilman

Submitted to the Department of Nuclear Science and Engineering on May 14, 2014 in Partial Fulfillment of the Requirements for the degree of

DOCTOR OF PHILOSOPHY
IN NUCLEAR SCIENCE AND ENGINEERING

ABSTRACT

Advanced modeling capabilities were developed for application to subcooled flow boiling through this work. The target was to introduce, and demonstrate, all necessary mechanisms required to accurately predict the temperature and heat flux for subcooled flow boiling in CFD simulations. The model was developed using an experiemntally based mechanistic approach, where the goal was to accurately capture all physical phenomena that affect heat transfer and occur at the heated surface to correctly predict surface temperatures.

The proposed model adopts a similar approach to the classical heat partitioning method, but captures additional boiling physical phenomena. It introduces 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. The convection term is modified to account for increased surface roughness caused by the presence of the bubbles on the heated surface. The quenching term accounts for bringing the bubble dry spot back to the wall superheat prior to bubble inception. In addition to the changes to these three classic components, a sliding conduction term is added to capture the increased heat transfer due to bubble sliding on the heated surface prior to lift-off. The sliding conduction component includes all heat removal associated with transient conduction caused by disruption of the thermal boundary layer. The new method extends the generality and applicability of boiling models in CFD through a fully mechanistic representation.

The new model also tracks the dry surface area during boiling for possible application in DNB predictions. A statistical tracking method for bubble location on the heated surface provides information on the bubble merging probability and prevents the active nucleation site density from reaching un-physical values. The model was implemented in the CFD software STAR-CCM+, and the wall temperature predictions were recorded and compared against the standard model's predictions and experimental data for a range of mass fluxes, heat fluxes, inlet subcoolings, and pressures. In general, the new model predicts wall temperatures closer to experimental data for both low and high pressures when compared against the standard model. The new model also converges at higher heat fluxes and greater subcoolings than the standard model.

Thesis Supervisor: Emilio Baglietto
Title: Assistant Professor of Nuclear Science and Engineering

Thesis Reader: Jacopo Buongiorno
Title: Associate Professor of Nuclear Science and Engineering

CASL-U-2014-0205-000

Thesis Reader: Evelyn N. Wang
Title: Associate Professor of Mechanical Engineering

Thesis Reader: Paolo Di Marco
Title: Associate Professor of Engineering Thermodynamics and Heat Transfer, School of Engineering, University of Pisa

ACKNOWLEDGEMENTS

I would first like to thank my thesis supervisor, Professor Emilio Baglietto, for all his support with my research. The completion of my Ph.D. would not have been possible without his guidance and patience in teaching me about the practices of CFD and modeling multiphase flow. He also spent many hours brainstorming with me to develop much of the new wall boiling model. I also greatly enjoyed having the chance to work under him and appreciate all of his professional advice and guidance. I would also like to thank Professor Jacopo Buongiorno for his support with both my research and academics. I have truly enjoyed working with him both in research and also in RTC. Additionally, I express thanks to both Professor Evelyn Wang and Professor Paolo Di Marco for their input with my project.

I would like to extend my personal gratitude to Andrew Splawski and Paco Ezquerra who helped immensely with the linkage of my new boiling model in STAR-CCM+. The successful implementation of the new model would not have been possible without their help in understanding how to define and call variables from the STAR-CCM+ software. Paco also helped in locating and explaining errors in variable definitions in my external subroutines. In addition, Eric Forrest provided much insight on experimental work through many interesting discussions about high pressure experiments, and the physical phenomena I needed to describe in my new model. He also helped me understand how to determine the usefulness of an experimental data set for validation.

I extend my deepest gratitude to my fiancé Bren Phillips. His patience with me was astonishing, and he was always available when I needed help; whether it be with boosting my confidence, or aiding in my research by sharing his thoughts based on his wealth of experience in the heat transfer area. His calm demeanor and constant encouragement was key for my successful completion of graduate school. He provided much of the experimental data that I required for validation of my research, and was also readily available to answer all questions I had about how the data was gathered and what the trends represented physically. His support in me and all that I do, is immeasurable.

I would also like to thank the DNSE administrators I have worked with over my years in the department. They have had endless patience in answering all my questions and quickly fixing any problems that I have had. In particular, I would like to thank Peter Brenton, who seemed able to solve any issue that I was having and answer any question I had, Lisa Magnano-Bleheen, Clare Egan, Heather Barry, and Rachel Batista. Also, I would like to thank Christine Sherratt, the librarian working with NSE, for her help in gathering obscure journal and conference papers and theses from other universities.

My research group-mates, Giancarlo Lenci, Etienne Demarly, Joe Fricano, Rosemary Sugrue, Sasha Tan-Torres, Gustavo Montoya, Nazar Lubchenko, and Ben Magolan, provided much support and help with running and debugging my CFD simulations. In particular, my office neighbor Giancarlo was always available to help with any computer problem I had, and also provided encouragement and support whenever I was in need of it. Giancarlo was also an amazing ANS Co-President to work with, and the job was infinitely more fun doing it with him. Etienne was my ally in determining and investigating the best high pressure experimental data to compare my new model against. Best of luck to Etienne in the continuation of this project as well. Also, Rosie provided much insight on bubble departure diameter experiments and was extremely helpful and patient in helping me understand her force-balance bubble departure model.

My friends have also been vital in providing me with the support I needed to succeed. In particular I would like to thank my officemate and roommate Brittany Guyer, who gave me support and guidance during all of my time at MIT. Additionally, Koroush Shirvan is always helpful with installing software and running codes, and bringing me awesome, surprise food deliveries! Also, I would like to thank the MIT Women's Volleyball Club members who I immensely enjoy playing volleyball with and always boost my confidence! Finally, I would also like to acknowledge my very close friends from Valpo, Alyssa, David and Rachel, who have been a constant source of support.

My family has always backed me in my pursuits and dreams, and for that I want to express my deep gratitude. None of my successes would have been possible without their help and belief in my abilities.

I especially would like to thank my Mom, Dad, Rachel, Chris, and Heather. Also, I want to thank my grandparents, Dick and Mavis Collins, who moved me to MIT all the way from South Bend, Indiana and also helped me in my undergraduate education. In addition, thank you to my grandparents, Arnold and Dee Gilman, for their encouragement in all of my pursuits. Finally, I express thanks to my Aunt Linda Collins, who gave me great advice in preparing for and choosing a graduate school, and also provided guidance in the selection of a probability method employed in my model.

I would also like to acknowledge my funding sources for this project, both from the Consortium of Advanced Simulation of Light Water Reactors (CASL) and the MIT-DNSE Gyftopolous Fellowship.

Contents

ABSTRACT 3

ACRONYMS 11

NOMENCLATURE 11

LIST OF FIGURES 21

LIST OF TABLES 23

1 Introduction 25

1.1 Motivation . . . 25

1.2 Objective . . . 25

CHAPTER ONE: BACKGROUND AND CFD IMPLEMENTATION 27

2 Physical Insight on Boiling 27

2.1 General Boiling Description . . . 27

2.2 Bubble Growth . . . 28

2.3 Subcooled Flow Boiling . . . 32

2.4 MIT Flow Boiling Facility . . . 34

2.5 MIT Flow Boiling Facility CFD Study . . . 38

2.5.1 Geometry . . . 39

2.5.2 Flow Analysis . . . 41

3 Current Wall Boiling Partitioning Approach 45

3.1 Eulerian Multi-Phase Framework in CFD . . . 45

3.2 Modeling of Turbulent Flows . . . 47

3.3 Current Partitioned Heat Flux Boiling Model . . . 48

3.4 Baseline Model Demonstration . . . 56

4 STAR-CCM+ CFD Software 59

4.1 STAR-CCM+ Capabilities . . . 59

4.2 Solution Algorithms . . . 59

CHAPTER TWO: NEW WALL BOILING MODEL DESCRIPTION 63

5 Nucleation Site Density Modification 63

6 Wall Boiling Model Formulation **67**

6.1 Forced Convection Component 67

6.2 Sliding Conduction Component 71

6.3 Quenching Component 75

6.4 Evaporation Component 76

7 Bubble Tracking, Merging, and Dry Surface Area **79**

8 Model Implementation in STAR-CCM+ Software **81**

8.1 User Code External Routine 81

8.2 Linkage to Software 81

8.3 Field Functions 82

9 Mechanistic Force-Balance Models **85**

9.1 Bubble Departure Diameter 85

9.2 Bubble Lift-off Diameter 88

9.3 Bubble Departure Frequency 88

CHAPTER THREE: TESTING AND VALIDATION OF NEW MODEL **91**

10 Model Validation **91**

10.1 Comparison with MIT Flow Boiling Facility Data 91

10.1.1 Wall Temperature Predictions at 1.05 Bar 92

10.1.2 Wall Temperature Predictions at 1.5 Bar 100

10.1.3 Wall Temperature Predictions at 2 Bar 102

10.2 Comparison with High Pressure Rohsenow Data 107

11 Model Sensitivity Studies **113**

11.1 Active Nucleation Site Density Models 113

11.2 Bubble Departure Frequency 115

11.3 Maximum Eötvös number 116

11.4 Grid Size 120

11.5 Bubble Dry Spot Temperature Change 121

11.6 Maximum Microlayer Thickness 122

12 Summary, Conclusions, and Future Work **125**

12.1 Summary 125

12.2 Conclusions 125

12.3 Suggested Future Work 127

Appendices **134**

Appendix A User Code Files **134**

Appendix B Field Function Code **188**

ACRONYMS

Notation	Description	Page List
CASL	Consortium for Advanced Simulation of Light Water Reactors	25, 125
CFD	Computational Fluid Dynamics	3, 25, 34, 48, 70, 125
CHF	Critical Heat Flux	25, 28, 30, 79, 125
DNB	Departure from Nucleate Boiling	3, 25, 28, 32, 125
DNS	Direct Numerical Simulation	47, 70
EMP	Eulerian Multi-Phase	45, 92
HSV	High-Speed Video	17, 30, 37
IR	Infrared	17, 29–31, 36, 91
ITO	Indium-Tin Oxide	34, 36
LES	Large Eddy Simulation	47, 70
ONB	Onset of Nucleate Boiling	17, 26, 33
OSV	Onset of Significant Voids	17, 26, 33, 34
RANS	Reynolds-Averaged Navier Stokes	47, 48
STAR-CCM+	CFD software by CD-Adapco	3, 59, 91, 125
URANS	Unsteady Reynolds-Averaged Navier Stokes	47
VERA	Virtual Environment for Reactor Analysis	25

NOMENCLATURE

Notation	Description	Page List
A_{ij}^D	Linearized drag coefficient	51
A_{dry}	Dry area	79
A_f	Flow area	64
A_h	Heater area	79
α	Advancing contact angle	85
α_k	Local volume fraction for the kth phase	45, 52
α_{dry}	Wall dryout breakpoint	55
α_v	Local volume fraction of the vapor phase	55
a_{sl}	Bubble sliding area	72
β	Receding contact angle	85
C_D	Drag coefficient	51
C_{kl}	Inter-phase heat-transfer term	47
C_L	Lift coefficient	55, 113
C_μ	Turbulent viscosity scaling coefficient	46, 53, 113
θ	Static contact angle	27, 63
c_{pl}	Specific heat of the liquid	48
c_{ph}	Specific heat of the heater	75
C^{VM}	Virtual mass coefficient	54
D_{avg}	Average sliding bubble diameter	74
D_{bo}	Initial bubble lift-off diameter as a function of time	49
D_d	Bubble departure diameter	34, 113

Notation	Description	Page List
δ_{max}	Maximum microlayer thickness	76
D_h	Hydraulic diameter	64
D_l	Bubble lift-off diameter	32, 71, 113
D_m	Bubble movement diameter	71
D_{sl}	Bubble sliding diameter	72
d_w	Bubble foot diameter	85
ε_k	Dissipation of turbulent energy of the kth phase	46
e_k	Specific internal energy	47
Π_H^k	Interfacial energy sources	47
H_k	Total enthalpy of the fluid	46
Eo	Eötvös number	51, 74, 113
η_l	Liquid thermal diffusivity	65
f	Bubble departure frequency	28, 34
F_A	Bubble influence area factor	48
F_b	Buoyancy force	85
F_{cp}	Contact pressure force	85
F_D^k	Interfacial force density	46
$\boldsymbol{F}_D^{k,drag}$	Interfacial drag force	46
$\boldsymbol{F}_D^{k,non-drag}$	Non-drag interfacial forces	46
F_{du}	Unsteady drag force	85
F_h	Hydrodynamic pressure force	85
Fo	Fourier number	49
$\boldsymbol{F}^{k,bodyforces}$	Body forces	46
F_{qs}	Quasi-steady drag force	85
F_{sL}	Shear lift force	85
F_{sx}	Surface tension force in the x-direaction	85
F_{sy}	Surface tension force in the y-direaction	85
f_x	Body force acting in the x-direction	46
f_y	Body force acting in the y-direction	46
f_z	Body force acting in the z-direction	46
G	Mass flux	64
$\boldsymbol{g}$	Gravity force	34
G_k	Production term due to gravity for kth phase	52
G_k^{NL}	Non-linear production term	52
G_s	Dimensionless shear rate	87
h_b	Internal liquid coefficient B	82
h_{fc}	Heat transfer coefficient for liquid forced convection	48, 70, 82
h_{fg}	Latent heat of evaporation	49
h_l	Liquid heat transfer coefficient for boiling	49
h_{sc}	Heat transfer coefficient for sliding conduction term	81
h_{vc}	Heat transfer coefficient for vapor convection	55
Γ_k	Interfacial mass source	45
Ja	Jakob number	49, 88
κ	Von Karman constant	68

Notation	Description	Page List
K_{dry}	Effective wall contact area fraction	55
k_k	Turbulent kinetic energy of the kth phase	46
k_l	Thermal conductivity of the liquid	48
k	Reference to the kth phase	45
K_{quench}	Area fraction that undergoes quenching	48
l	Bubble sliding length	72
λ_k	Thermal conductivity of the kth phase	47
λ'	Cavity length scale	50
$\dot{m}_{kl}$	Mass transfer from the kth phase to the lth phase	45
$\dot{m}_{lk}$	Mass transfer from the lth phase to the kth phase	45
Mo	Morton number	74
$\mathbf{\Omega}_k$	Interfacial momentum sources	46
μ_k	Dynamic viscosity of the kth phase	46
$\mu_{k,T}$	Turbulent viscosity of the kth phase	46
μ_l	Dynamic viscosity of the liquid	49
N''	Active nucleation site density	34, 82
N_{merged}	Number of bubbles a sliding bubble merged with	74
N''^*	Reduced nucleation site density from sliding bubbles	74
Nu	Nusselt number	49
ν_l	Kinematic viscosity of the liquid	86
P_h	Heated perimeter	64
Φ_H^k	Effects of viscous extra stresses	46
ϕ	Inclination angle of the bubble	33, 85
ϕ_e	Flux at the surface of a control volume	60
P_k	Shear production term for the kth phase	52
P_l	Liquid pressure	27
p	System pressure	46
p_k	Pressure of the kth phase	46
$Pr_{k,T}$	Turbulent Prandtl number	47
$\mathrm{Pr_l}$	Liquid Prandtl number	49
P_v	Vapor pressure	27
$\boldsymbol{q}$	Heat addition	47
q''_e	Evaporation component	49, 67
$q''_{e,init}$	Evaporation component from the initial bubble inception	76
$q''_{e,mL}$	Evaporation component from the microlayer evaporation	76
q''_{fc}	Forced convection component	48, 67
Q_H^{int}	Interfacial heat source	47
q''_q	Quenching component	48, 67
q''_{sc}	Sliding conduction component	67
q''_{tot}	Total heat flux applied to a wall	48, 67
q''_w	Wall heat flux	81
R	Instantaneous bubble radius	86
r	Equivalent sand-grain roughness height	68, 82
R_a	Average roughness height	69

Notation	Description	Page List
R_c	Minimum cavity radius	50
Re	Reynolds number	74
$\mathrm{Re_b}$	Bubble Reynolds number	49, 72, 86
Re_τ	Shear Reynolds number	70
$\mathrm{Re_{TP}}$	Two-phase Reynolds number	64
R_f	Nucleation site density reduction factor	74
R_g	Gas constant	50
ρ_h	Heater density	75
ρ_l	Liquid density	48
ρ_v	Vapor density	49
R^+	Roughness parameter	68
r_r	Radius of curvature of the bubble	87
r^*	Equilibrium bubble radius	27
$\dot{R}$	First time derivative of the bubble growth rate	87
$\ddot{R}$	Second time derivative of the bubble growth rate	87
s	Bubble spacing	69, 72
S_{dry}	Fraction of dry surface area	79, 82
$S_{\varepsilon 2}^{int}$	Turbulent dissipation source term	52
σ	Surface tension	27
S_{k2}^{int}	Turbulent kinetic energy source term	52
S_{k_l}	Particle induced turbulent kinetic energy source term	54
τ_w	Wall shear stress	68
$\boldsymbol{\tau}_k$	Stress tensor	46
T	Averaging timescale	45, 47
τ	Bubble period	28, 36
$\boldsymbol{\tau}_{k''}$	Linear relationship of the Reynolds stress to the averaged velocity	46
T_{bulk}	Bulk temperature	32
t_g	Bubble growth time in the bubble ebullition cycle	28, 65
ΔT_h	Temperature difference between bubble hot spot and average heater temperature	75
T_{in}	Inlet temperature	32
T_k	Temperature of the kth phase	47
T_l	Temperature of the lth phase	47
T_{sat}	Saturation temperature	27, 32
t_{sl}	Bubble sliding time	72
t^*	Time interval transient conduction occurs	72
ΔT_{sub}	Liquid subcooling	34
ΔT_{sup}	Wall superheat	27
t_w	Bubble wait time in the bubble ebullition cycle	28, 48
T_w	Wall temperature	27, 71
U	Flow velocity	86
$U_{b,rel}$	Relative velocity at the bubble center	49
$U_{b,y}$	Instantaneous velocity in the normal (y) direction	49

Notation	Description	Page List
$U_{b,z}$	Instantaneous velocity in the axial (z) direction	49
U_l	Bulk liquid velocity in the axial direction	49
$\boldsymbol{U}_l$	Liquid velocity vector	54
u^+	Non-dimensional velocity	68
$\boldsymbol{U}_r$	Slip velocity vector	54
$\boldsymbol{U}_v$	Vapor velocity vector	54
V	Volume based on an averaging length scale	45
ϑ	Angle of test section with respect to horizontal	32, 85
υ_b	Sliding bubble velocity	73
u_τ	Friction velocity	67
$\boldsymbol{U}_k$	Velocity vector for the kth phase	45
$\boldsymbol{U}_l$	Velocity vector for the lth phase	46
u_k	Velocity in x-direction	45
v_k	Velocity in y-direction	45
w_k	Velocity in z-direction	45
$\boldsymbol{v}_i$	Velocity vector of phase i	51
$\boldsymbol{v}_j$	Velocity vector of phase j	51
V_{mL}	Volume of the microlayer	76
V_q	Volume of the heater hot spot	75
x	Vapor Quality	64
y^+	Non-dimensional distance from wall	68

List of Figures

1.1 Illustration of a subcooled flow boiling channel indicating where the onset of nucleate boiling (ONB) and the onset of significant voids (OSV) occurs. 26

2.1 Illustration of the static contact angle of a bubble. 27

2.2 Illustration of the microlayer of a bubble formed at a nucleation site of a heated surface. . . 29

2.3 Example of the interference rings caused by the presence of the liquid microlayer and the dry center of the bubble base. Picture provided by Bren Phillips. 30

2.4 HSV for a single bubble life-cycle for frames 46-55 in water at q″ = 60 kW/m^2 with a time interval between images of 2.08 ms. Adapted from [14]. 30

2.5 IR data of the heater surface temperature for a single bubble life-cycle for frames 46-55 in water at q″ = 60 kW/m^2 with a time interval between images of 2.08 ms. Adapted from [14]. 31

2.6 Comparison of bubble, cold spot, and dry spot radii with corresponding predictions for water at q″ = 60 kW/m^2 with an error for measurments of $\pm 10\%$. Adapted from [14]. 31

2.7 Depiction of a bubble during the thermally-controlled growth period. 32

2.8 Pictures 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. 33

2.9 Diagram of the flow boiling loop. Courtesy of Dr. Greg DeWitt. 35

2.10 Drawing of the entrance and quartz regions provided by Bren Phillips. 36

2.11 Bubble departure diameters versus mass flux for 1.05 bar and 10°C subcooling. Adapted from [44]. 36

2.12 Active nucleation site densities for 1.05 bar and 10°C subcooling. Adapted from [44]. 37

2.13 HSV (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). Adapted from [44]. 37

2.14 The bubble sliding velocities versus the bulk fluid velocity at 1.05 bar. Adapted from [44]. . 38

2.15 The bubble period versus wall superheat at 1.05 bar and 10°C subcooling. Adapted from [44]. 38

2.16 Original CAD geometry of quartz cell region. 39

2.17 Extruded geometry to model the entire length of the channel. 40

2.18 Surface mesh at the location of the heater. 40

2.19 Cross-section of the volume mesh at the center of the heater perpendicular to the flow. . . . 41

2.20 Wall y^+ values in the heater region. 41

2.21 Secondary flow velocities with an initial inlet velocity of 2.0 m/s in the +z direction. 42

2.22 Flow direction analysis near the heater to determine the influence of the outlet geometry. . . 43

2.23 Comparison of the velocity profile at the location at the center of the heater for a constant inlet velocity and an inlet velocity that varies in the x-direction. 43

3.1 Depiction of the original heat flux partitioning for subcooled flow boiling. 48

3.2 CFD grid of spacer with mixing vanes [64]. 57

3.3 Rod and spacer contacts included in the model [64]. 57

3.4 Void fraction downstream of a mixing vane spacer [64]. 58

4.1 A typical CV layout with notation used for a 2-D Cartesian grid [50]. 60

4.2 The SIMPLE scheme used in an iterative flow solver for single-phase [55]. 62

5.1 Comparison of the Lemmert-Chawla, Kocamustafaogullari-Ishii, and Hibiki-Ishii nucleation site density models at 1.05 bar. 64

5.2 Comparison of the Hibiki-Ishii model at three different pressures to the Lemmert-Chawla model. 65

5.3 Illustration of a nucleation site that is underneath a bubble already present on the heated surface. 65

5.4 Illustration of the effect of the probability method on the nucleation site density. 66

6.1 Depiction of the heat flux partitioning for subcooled flow boiling. 67

6.2 Velocity profile near the wall. 68

6.3 Depiction of the unit cell and volume taken up by a hemispherical bubble. 69

6.4 ITM Test Case 1 computational domain and illustration of the hemispherical obstacles [84]. 70

6.5 Velocity profile near the wall of the rough wall function (blue) compared to the DNS data of the smooth (red) and hemispherical obstacle (yellow) surfaces. 71

6.6 Illustration of bubble growth at a nucleation site until it departs from the site, slides, and lifts-off the surface. 72

6.7 Illustration of a bubble sliding from its inception point and then departing from the heater after sliding a distance given by l. 73

6.8 Illustration of a bubble sliding from its inception point and merging with another bubble after sliding a distance s. 73

6.9 Illustration of bubble growth on a heater and the area of the bubble that is involved in the quenching heat flux component. 76

6.10 Illustration of microlayer evaporation and condensation on a bubble attached to a heated wall. 76

6.11 Depiction of the microlayer showing the maximum thickness. 77

7.1 Illustration of the maximum distance between two bubble centers that allows for bubble merging. 79

7.2 Illustration of two bubbles circled in red that would merge on the heated surface. 80

9.1 Illustration of the forces acting on a bubble in subcooled flow boiling. Figure provided by Rosemary Sugrue. 86

9.2 Illustration of the forces acting on a bubble in subcooled flow boiling. 89

10.1 Original CAD geometry of the quartz cell region. 91

10.2 Full geometry (after extrusion) showing the vertical upflow position. 91

10.3 Cross-section of the volume mesh at the center of the heater for the MIT Flow Boiling Facility test cases that have a mass flux of 500 kg/m^2-s. 93

10.4 Wall y^+ values on the heater region for the MIT Flow Boiling test cases that have a mass flux of 500 kg/m^2-s. 94

10.5 Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 500 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 1.05 bar. . . . 94

10.6 The fraction of the total heat flux removed by each component in the new model's heat partitioning model at a mass flux of 500 kg/m^2-s, 10 K subcooling, and a pressure of 1.05 bar. 96

10.7 Cross-section of the volume mesh at the center of the heater for the MIT Flow Boiling Facility test cases that have a mass flux of 1000 kg/m^2-s. . . . 96

10.8 Wall y$^+$ values on the heater region for the MIT Flow Boiling test cases that have a mass flux of 1000 kg/m^2-s. . . . 97

10.9 Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1000 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 1.05 bar. . . . 97

10.10 Cross-section of the volume mesh at the center of the heater for the MIT Flow Boiling Facility test cases that have a mass flux of 1250 kg/m^2-s. . . . 98

10.11 Wall y$^+$ values on the heater region for the MIT Flow Boiling test case with a mass flux of 1250 kg/m^2-s. . . . 99

10.12 Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1250 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 1.05 bar. . . . 99

10.13 Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 500 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 1.5 bar. . . . 100

10.14 The fraction of the total heat flux removed by each component in the new model's heat partitioning model at a mass flux of 500 kg/m^2-s, 10 K subcooling, and a pressure of 1.5 bar. 101

10.15 Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1000 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 1.5 bar. . . . 101

10.16 Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1250 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 1.5 bar. . . . 102

10.17 Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 500 kg/m^2-s, inlet subcooling of 5 K, and at a pressure of 2.0 bar. . . . 103

10.18 Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 500 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 2.0 bar. . . . 103

10.19 Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 500 kg/m^2-s, inlet subcooling of 15 K, and at a pressure of 2.0 bar. . . . 104

10.20 Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1000 kg/m^2-s, inlet subcooling of 5 K, and at a pressure of 2.0 bar. . . . 104

10.21 Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1000 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 2.0 bar. . . . 105

10.22 Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1000 kg/m^2-s, inlet subcooling of 15 K, and at a pressure of 2.0 bar. 105

10.23 Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1250 kg/m^2-s, inlet subcooling of 5 K, and at a pressure of 2.0 bar. 106

10.24 Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1250 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 2.0 bar. 106

10.25 Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1250 kg/m^2-s, inlet subcooling of 15 K, and at a pressure of 2.0 bar. 107

10.26 The locations of the outer wall temperature measurements 2, 3, 4, 5, and 6 for the Rohsenow test data. 108

10.27 The Rohsenow test section original CAD geometry. 108

10.28 The extruded Rohsenow test case geometry near the outlet of the tube. 109

10.29 A cross-section view of the meshed Rohsenow tube geometry. 109

10.30 The wall y^+ values of the Rohsenow test section for an inlet velocity of 3.058 m/s. 110

10.31 The fraction of the total heat flux removed by each component in the new model's heat partitioning model for the Rohsenow test case with heat flux of 5.11 MW/m^2 at a pressure of 2000 psi. 111

11.1 Boiling curves calculated using the standard set of models and compared to the experimental data for a mass flux of 500 kg/m^2-s, a pressure of 1.05 bar, and an inlet subcooling of 10 K. 114

11.2 Boiling curves calculated using the new wall boiling model and compared to the experimental data for a mass flux of 500 kg/m^2-s, a pressure of 1.05 bar, and an inlet subcooling of 10 K. 114

11.3 Wall temperature predictions calculated using the new wall boiling model with varying bubble departure frequencies, and compared to the experimental data for a mass flux of 500 kg/m^2-s, a pressure of 2.0 bar, and an inlet subcooling of 10 K. 115

11.4 Wall temperature predictions calculated using the new wall boiling model with varying bubble departure frequencies, and compared to the experimental data for a mass flux of 500 kg/m^2-s, a pressure of 1.5 bar, and an inlet subcooling of 10 K. 116

11.5 Average heater temperature predictions for the experiment and the new model with varied maximum Eo numbers for a mass flux of 500 kg/m^2-s, inlet subcooling of 15 K, and pressure of 2.0 bar. 117

11.6 Average heater temperature predictions for the experiment and the new model with varied maximum Eo numbers for a mass flux of 1000 kg/m^2-s, inlet subcooling of 15 K, and pressure of 2.0 bar. 117

11.7 Average heater temperature predictions for the experiment and the new model with varied maximum Eo numbers for a mass flux of 500 kg/m^2-s, inlet subcooling of 10 K, and pressure of 2.0 bar. 118

11.8 Average heater temperature predictions for the experiment and the new model with varied maximum Eo numbers for a mass flux of 1000 kg/m^2-s, inlet subcooling of 10 K, and pressure of 2.0 bar. 118

11.9 Average heater temperature predictions for the experiment and the new model with varied maximum Eo numbers for a mass flux of 500 kg/m^2-s, inlet subcooling of 5 K, and pressure of 2.0 bar. 119

11.10 Average heater temperature predictions for the experiment and the new model with varied maximum Eo numbers for a mass flux of 1000 kg/m^2-s, inlet subcooling of 5 K, and pressure of 2.0 bar. 119

11.11 Average heater temperature predictions for the experiment and the new model with varied maximum Eo numbers for a mass flux of 500 kg/m^2-s, inlet subcooling of 10 K, and pressure of 1.5 bar. 120

11.12 Average heater temperature predictions for the experiment and the new model with varied maximum Eo numbers for a mass flux of 1000 kg/m^2-s, inlet subcooling of 10 K, and pressure of 1.5 bar. 120

11.13 Wall temperature predictions calculated using the new wall boiling model with varying bubble hot spot temperatures, and compared to the experimental data for a mass flux of 500 kg/m^2-s, a pressure of 2.0 bar, and an inlet subcooling of 10 K. 122

11.14 Wall temperature predictions calculated using the new wall boiling model with varying bubble hot spot temperatures, and compared to the experimental data for a mass flux of 500 kg/m^2-s, a pressure of 1.5 bar, and an inlet subcooling of 10 K. 122

11.15 Wall temperature predictions calculated using the new wall boiling model with varying maximum microlayer thickness, and compared to the experimental data for a mass flux of 500 kg/m^2-s, a pressure of 2.0 bar, and an inlet subcooling of 10 K. 123

11.16 Wall temperature predictions calculated using the new wall boiling model with varying maximum microlayer thickness, and compared to the experimental data for a mass flux of 500 kg/m^2-s, a pressure of 1.5 bar, and an inlet subcooling of 10 K. 124

List of Tables

2.1 Potential experimental conditions for the MIT Flow Boiling Facility. 35
2.2 Extrusion parameters from the original CAD geometry. 39
2.3 Meshing values used for building the reference base case model. 40
2.4 Physics continuum initial conditions for the reference base case. 41
3.1 Constants used in the standard k-ε model. 52
3.2 Non-linear coefficients. 54
3.3 Quadratic coefficients. 54
3.4 Cubic coefficients. 54
3.5 Standard Wall Boiling Models 56
6.1 Definition of variables used in the smooth wall functions. 68
10.1 Meshing values used for the MIT Flow Boiling Facility test cases that have a mass flux of 500 kg/m^2-s. 93
10.2 The fraction of dry surface area for a mass flux of 500 kg/m^2-s, inlet subcooling of 10 K, and pressure of 1.05 bar. 95
10.3 Meshing values used for the MIT Flow Boiling Facility test cases that have a mass flux of 1000 kg/m^2-s. 96
10.4 Meshing values used for the MIT Flow Boiling Facility test cases that have a mass flux of 1250 kg/m^2-s. 98
10.5 Meshing values used for the Rohsenow test cases that have an inlet velocity of 3.048 m/s. . . 109
10.6 Inlet temperature for the Rohsenow test cases at 2000 psi and an inlet velocity of 3.048 m/s. . 110
10.7 Inlet temperature for the Rohsenow test cases at 2000 psi and an inlet velocity of 3.048 m/s. . 111
11.1 Parameters used in full sensitivity study [36]. 113
11.2 Liquid heat transfer coefficient and wall superheat predictions with varied base cell size. . . 121
11.3 Liquid heat transfer coefficient and wall superheat calculated with varied prism layer thickness. 121

1 Introduction

1.1 Motivation

The need to optimize flow conditions for new and improved heat transfer designs has driven studies in numerical analyses, such as Computational Fluid Dynamics (CFD), to supplement the use of experiments. In many industrial applications, subcooled flow boiling is used when a high heat transfer coefficient is desired. Often in these applications, it is important to know the wall temperature of the heated material because the temperature affects not only the structural integrity of the material, but also the corrosion and corrosion product deposition rates on the surface.

The existing CFD flow boiling models do not incorporate all the physical phenomena that occur on a heated surface, but those details are necessary to accurately predict the wall temperature. Also, existing models have been validated and tweaked to correctly predict the void fraction distribution in channels, rather than capture the correct heat flux partitioning from the heated wall to the fluid that is necessary for the calculation of the wall temperature.

In addition, the Consortium for Advanced Simulation of Light Water Reactors (CASL) advances simulation technology by connecting government, academia, and industry for technology development and fundamental research. The aim is to have a Virtual Environment for Reactor Analysis (VERA) to be able to assess key nuclear energy industry challenges. VERA will predict reactor performance through simulation in order to enhance safety, economics, and reliability [1]. An important contribution of this work is to develop a wall boiling model that can also provide insight to the critical heat flux (CHF) for a given system. CHF occurs when the thermal limit of a a system is reached such that the heat transfer coefficient from the surface to the fluid suddenly decreases, which can cause overheating of the surface. To correctly predict the wall temperatures and describe CHF using CFD in VERA, a complete understanding of the complex boiling phenomena and boiling mechanisms must be assembled.

1.2 Objective

Subcooled flow boiling is used as the method of heat removal from the fuel assemblies in Light Water Reactors (LWRs), depicted in Figure 1.1, and it is a popular method to use because it has a high heat transfer coefficient. Subcooled flow boiling occurs when the inlet temperature of a heated channel is below the saturation temperature. As heat is added to the liquid through contact with the heated walls of the channel, boiling can occur where the liquid is at a temperature greater than the saturation temperature even though the bulk temperature may remain subcooled. In order to maintain this efficient heat transfer regime, CHF, which is dependent on the geometry and flow conditions [2], must be avoided.

Here, the objective is to develop a more general, and local, boiling closure model for subcooled flow boiling conditions. The model includes all physical boiling phenomena and necessary mechanisms to accurately predict the temperature and heat flux at the wall in CFD simulations for subcooled flow boiling. The model explicitly tracks the dry surface area during boiling so that the model can be further extended to predict departure from nucleate boiling (DNB) by studying the fraction of dry surface area as the heat flux increases for a system. DNB describes the phenomenon when the heat transfer dramatically decreases because a partial vapor film forms at the wall and nucleate boiling no longer occurs [2].

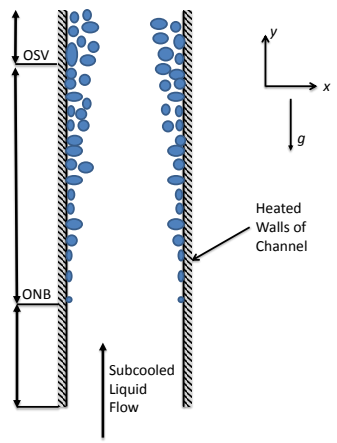


Figure 1.1. Illustration of a subcooled flow boiling channel indicating where the onset of nucleate boiling (ONB) and the onset of significant voids (OSV) occurs.

CHAPTER ONE: BACKGROUND AND CFD IMPLEMENTATION

2 Physical Insight on Boiling

2.1 General Boiling Description

Saturation conditions, at a given pressure, refer to the properties of a fluid at a temperature where both vapor and liquid can coexist. The temperature at which this occurs is called the saturation temperature and is denoted as T_{sat}. Vaporization of a liquid can occur when heat is transferred through a solid wall of a structure containing the liquid. When liquid is heated in this way, the liquid in the region closest to the wall has the highest temperature and a thermal boundary layer is formed. The thermal layer develops from both convection and transient conduction caused by bubble movement on the wall [3].

When enough heat is added to the system, the liquid in contact with the wall may reach and/or exceed the equilibrium saturation temperature. If the nucleation temperature required for the conditions of the system is reached, evaporation takes place at the solid-liquid interface by the formation of a vapor bubble. This type of boiling is often referred to as heterogeneous nucleation because the vapor nucleates from specific points on the heated surface known as nucleation sites [3, 4].

The initial nucleation of vapor typically occurs at a cavity or crevice on the heated surface that traps gas in the solid. Real surfaces have irregularities such as pits, scratches, or other imperfections that can vary greatly in size. Many of these cavities contain entrapped gas and vaporization of the liquid occurs at the liquid-gas interface in the cavity. For the vapor embryo to grow past the mouth of a cavity for a given contact angle and radius of the cavity, the wall superheat (ΔT_{sup}) must exceed the equilibrium value for the minimum interface radius. The wall superheat is defined as the difference between the wall temperature (T_w) and the saturation temperature [3, 5].

The contact angle of the interface of the vapor embryo on a surface affects the shape and formation of the vapor. The contact angle is illustrated in Figure 2.1 and is denoted as θ. The contact angle is dependent on the surface chemistry between a given surface and liquid [3]. Highly wetting fluids have a small contact angle, while non-wetting fluids have a larger (>90°) contact angle.

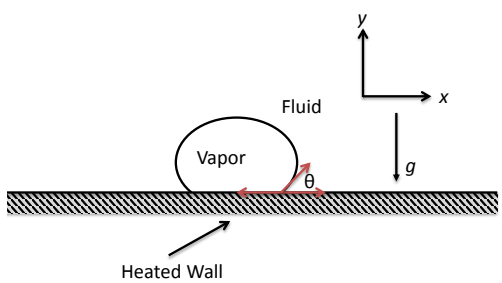


Figure 2.1. Illustration of the static contact angle of a bubble.

The interfacial tension, also referred to as surface tension (σ), between the liquid and vapor phases also greatly affects the shape and size of the vapor. The pressure difference across the vapor-liquid interface is related to the surface tension and the interface geometry. The mechanical equilibrium that is required for bubble formation is shown in (2.1), where P_v and P_l are the vapor and liquid pressures respectively. Assuming a spherical bubble shape, the bubble equilibrium radius is r^* [6]. A more general equation is shown in (2.2), where R_1 and R_2 are the radii of curvature of the interface for the two directions parallel to the surface. During the process of bubble growth, the interfacial tension along the triple contact line (the location where the vapor, liquid, and solid meet) tends to hold the bubble to the solid surface. The forces

from the fluid surrounding the vapor, such as buoyancy, drag, and lift, tend to force the bubble to depart from the surface, except when the bubble grows on a downward facing surface. In this case, the buoyancy force will tend to keep the bubble attached to the wall. The forces responsible for bubble detachment increase as the bubble grows larger. The surface characteristics, such as roughness, also affects the shape of the vapor forming on the surface [2, 3].

$$P_v - P_l = \frac{2\sigma}{r^*} \tag{2.1}$$

$$P_v - P_l = \sigma \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \tag{2.2}$$

The ebullition cycle describes the growth and release of vapor bubbles from an active nucleation site. After the initial bubble forms and releases off the surface, cooler bulk liquid comes in contact with the surface where the bubble was previously attached. This new liquid adjacent to the wall is heated to reform the thermal boundary layer near the wall. The waiting time (t_w) describes the time after the bubble departs from its site and until the next bubble forms at the same nucleation site. The time during which a bubble is growing and attached to its nucleation site is denoted as t_g. The bubble period (τ) is then calculated as shown in (2.3), and the bubble departure frequency (f) is the inverse of the bubble period [3].

$$\tau = t_g + t_w \tag{2.3}$$

The phase change of the liquid coolant to a vapor from the transfer of heat from the heated surface is a highly efficient mode of heat transfer. The limiting condition to this heat transfer method is referred to as CHF, the boiling crisis, or DNB. The change from the efficient heat transfer regime to the boiling crisis may occur on a short timescale (less than a second), and it occurs when the liquid no longer comes into direct contact with the heated surface. This occurs because a vapor film forms on the surface, separating the heated wall from the liquid. Without the direct contact of liquid with the surface, the heat transfer rate drops significantly because the vapor blanket acts as a barrier to heat flow and the temperature of the surface suddenly increases. This can cause serious problems with the integrity of the heated material, such as a physical burnout of the material that can lead to a catastrophic failure of the device [7].

There boiling system types- pool boiling and flow boiling. Pool boiling refers to boiling on a heated surface that is immersed in a pool of initially stagnant liquid. Flow boiling is when boiling occurs on a heated surface that has a liquid flowing over it at some mass flow rate [2]. The focus of this work is on the phenomena that occur during flow boiling, but the model is built such that it could also be used in pool boiling cases.

2.2 Bubble Growth

In the initial stage of bubble growth, the liquid in contact with the interface is highly superheated. When the bubble emerges from the nucleation site cavity, the sudden increase in the radius of curvature of the bubble causes a rapid expansion of the bubble. The inertia of the liquid resists the rapid growth and the bubble grows in an approximately hemispherical shape. This initial, high evaporation rate bubble growth is referred to as inertia-controlled growth.

As the bubble quickly grows in the radial direction, a thin layer of liquid is trapped beneath the lower portion of the bubble interface against the heated wall. This layer is called the microlayer and is depicted in Figure 2.2 [3]. The first experimental confirmation of the microlayer was done by Cooper and Lloyd [8].

After the initial formation of the bubble and microlayer, the bubble may continue to grow as heat is transferred by evaporation of the microlayer, and the bubble tends to become more spherical in shape. This

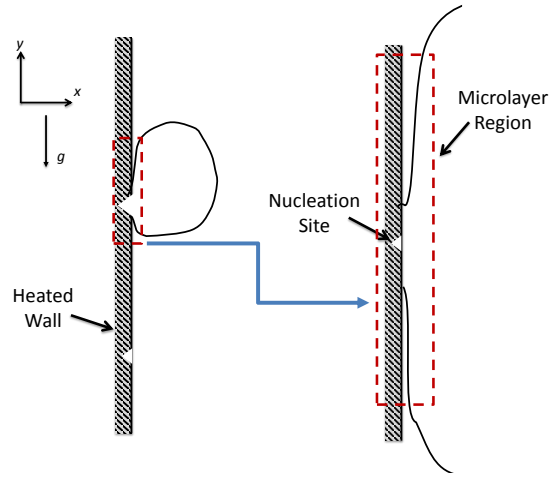


Figure 2.2. Illustration of the microlayer of a bubble formed at a nucleation site of a heated surface.

slower bubble growth state is referred to as thermally-controlled growth. Evaporation can also occur from the adjacent superheated liquid in contact with the bubble. The microlayer can completely evaporate near the nucleation site. This causes a small dry spot on the heated surface that has an elevated temperature due to a less efficient heat transfer where vapor is in contact with the wall. During boiling, this small area around the nucleation site will have strong surface temperature fluctuations as the bubble growth initiates, the microlayer evaporates, and the bubble departs from the surface allowing cooler liquid to come in contact with the surface [3, 9].

The characteristics of a system indicate whether the bubble growth is typically inertia-controlled or thermally-controlled. The inertia-controlled growth can govern the departure size for very small bubbles and this causes the bubbles to depart immediately from the surface after the rapid expansion from the nucleation site. This occurs in systems that have high wall superheat, high heat flux, very smooth and polished surfaces, highly wetting liquids (low contact angle), low latent heat of vaporization, or low system pressures because this typically has low vapor density. Also, if the system has a high mass flux, the bubble tends to depart early in the cycle. Systems that have the bubble departure size governed by the thermally-controlled growth stage are characterized by low wall superheat, low heat flux, rough surfaces, high latent heat of vaporization, or relatively high system pressure [10].

The new boiling model is built to be applicable for systems of a variety of conditions, and it must therefore include the phenomena that occur during both the intertia-controlled and thermally-controlled growth stages. As a result, the microlayer evaporation is included in the model.

Infrared (IR) thermography has been used with IR transparent heaters, typically made of an optical grade silicon wafer [11, 12], to provide information on the phenomena that occurs while a bubble is attached to a heated wall. 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 lighter in wetted areas.

The IR images provide detailed data on the features of a growing bubble. The difficult challenge is to distinguish the dry surface from the thin liquid microlayer that is only a few microns thick while the microlayer may be rapidly evaporating [9]. For IR transparent heaters, the detection and measurement of the thickness of the microlayer can be completed through the analysis of the interference fringe patterns that occur from the IR light passing through the thin liquid films. This method also detects the dark dry spot that is in the center of the bubble and is shown in Figure 2.3. The spacing of the rings in Figure 2.3 are used to reconstruct the shape of the microlayer over time.

IR opaque substrates have also been used on the heater to capture information about the microlayer by

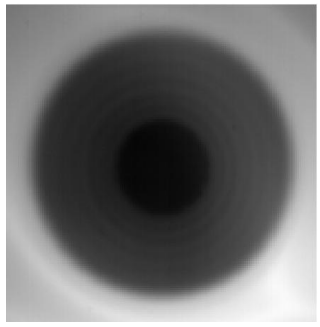


Figure 2.3. Example of the interference rings caused by the presence of the liquid microlayer and the dry center of the bubble base. Picture provided by Bren Phillips.

measuring the temperature profile of the surface with very high spatial resolution [13, 14]. When using IR opaque heaters, the temperature of the heated surface is measured. Therefore, the lighter colored areas represent the warmer area and the dark area is the cooler area. The measurement of the evaporated microlayer radius can thus be calculated through the measurement of the expanding hotter (dry) area under a bubble during thermally-controlled bubble growth. A single bubble cycle in pool boiling was investigated by Gerardi [14] using both high-speed video (HSV) and IR thermography on an IR opaque heater. The HSV images are shown in Figure 2.4 and the IR images used to measure the radius of the dryout area radius are shown in Figure 2.5.

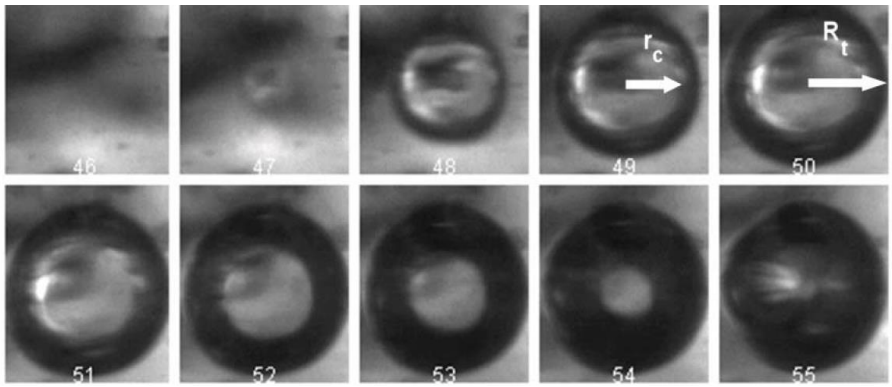


Figure 2.4. HSV for a single bubble life-cycle for frames 46-55 in water at q″ = 60 kW/m^2 with a time interval between images of 2.08 ms. Adapted from [14].

The expansion of the bubble, dry area, and influence area (cold area) of the bubble were measured and are shown in Figure 2.6. The dry area is used to calculate the inner hot spot radius of the bubble, which corresponds to the radius of evaporated microlayer.

Additional studies have also noted the importance of microlayer evaporation in predicting heat fluxes from nucleate boiling [8, 15–17] with Zhao et al. [18] developing a dynamic microlayer model to predict heat fluxes in fully developed nucleate boiling regions (including CHF). At high heat fluxes, the IR fringe pattern 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 [11].

When bubble growth is thermally-controlled, the top of the bubble may protrude into the subcooled region of the thermal layer near the wall. When this occurs, condensation at the top of the bubble competes with the evaporation from the microlayer and superheated region for the bubble growth and this is depicted in Figure 2.7.

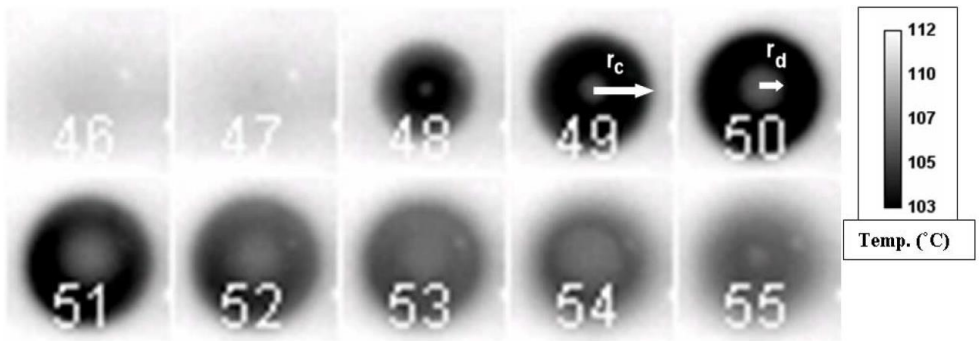


Figure 2.5. IR data of the heater surface temperature for a single bubble life-cycle for frames 46-55 in water at q″ = 60 kW/m^2 with a time interval between images of 2.08 ms. Adapted from [14].

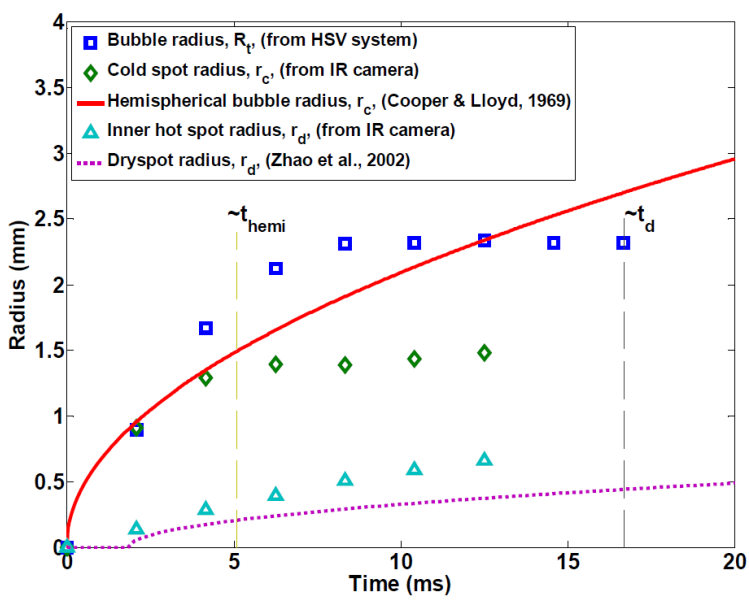


Figure 2.6. Comparison of bubble, cold spot, and dry spot radii with corresponding predictions for water at q″ = 60 kW/m^2 with an error for measurments of ±10%. Adapted from [14].

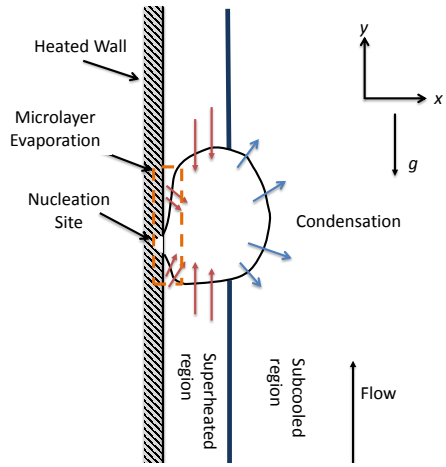


Figure 2.7. Depiction of a bubble during the thermally-controlled growth period.

2.3 Subcooled Flow Boiling

Turbulent flow in a vertical upflow channel with an axially uniform heat addition has multiple heat transfer regimes. First, the fluid enters the channel with a temperature (T_{in}) below T_{sat}. The liquid is initially heated by forced convection until the liquid near the wall has a great enough wall superheat that nucleate boiling occurs. The boiling process and additional turbulence caused by the bubble movement enhances the heat transfer. The bulk liquid temperature (T_{bulk}) will continue to rise and if it reaches T_{sat}, it is called saturated nucleate boiling. In this regime, bubbles can flow and merge in the bulk flow which can change the flow patterns. As boiling continues, the vapor can form as a core in the channel center with a thin film of liquid on the walls and small droplets in the vapor core. If the liquid film is depleted due to evaporation or liquid entrainment, the condition is known as dryout. In this regime, the vapor heat transfer is less efficient than the liquid heat transfer and the wall temperature can rise quickly. A high vapor generation rate can cause DNB to occur before dryout by establishing a vapor film that separates the liquid from the wall [2, 19].

In subcooled flow boiling, bubbles exhibit different behavior depending on the flow regime. In particular, the bubble lift-off diameter (D_l) from the heater surface is strongly dependent on wall superheat with a slight dependency on flow velocity. This observation is different from the bubble departure diameter (bubble departure from the active nucleation site, not necessarily lifting-off the heated surface), which is strongly dependent on both the wall superheat and liquid 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 near the wall [20]. The bubbles can form and also 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 when the liquid is highly subcooled [15, 21].

A recent study by Sugrue et al. [22, 23] was conducted to study the bubble departure diameters over a range of pressures, mass fluxes, heat fluxes, subcoolings, and inclination angle (ϑ) of the test section. The investigation indicated that as the inclination angle moves from vertical upflow to a 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. [24]. 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 Lin et al. [25] 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 [26].

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, it also has a slight angle of inclination (ϕ) caused by the asymmetrical bubble growth that occurs in flow boiling [15, 27]. 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 [20, 27]. In this type of flow regime, bubble sliding can be considered the main mode of heat transfer [28, 29]. Both microlayer evaporation 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 [29].

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 [28].

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 [28]. Other photographic studies of subcooled flow boiling have shown that bubbles can grow significantly while sliding before lift-off [30, 31]. Time snapshots of a growing and sliding bubble are shown in Figure 2.8.

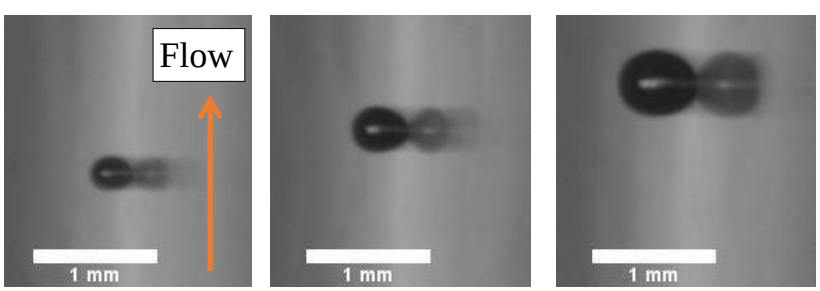


Figure 2.8. Pictures of a bubble sliding captured at the MIT Flow Boiling Facility by Bren Phillips. Test was completed using water at atmospheric pressure, heat flux of 130 kW/m^2, and 10°C subcooling.

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 [28]. The heat transfer contribution of sliding bubbles decreases as the heat flux increases because of bubble interaction on the heater [29].

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. Quenching refers to the process of transient conduction when cooler liquid comes into contact with the heated surface after a bubble departs from the surface.

Additionally at high heat fluxes, the dependence on the surface quantities, such as the nucleation site density, may no longer valid [33]. In the region of OSV, the bubbles have such a high degree of interaction that single bubbles can no longer be tracked by experimental techniques [28].

Comparisons between CFD predictions and experimental data [33–35] have illustrated the high sensitivity of current boiling closure parameters such as bubble departure diameter (D_d), bubble departure frequency (f), and active nucleation site density (N'') on factors such as the wall superheat. In particular, an in-depth study on numerous parameters used in multiphase CFD modeling was completed to determine what the results are most sensitive to [36] and is described in more detail in Section 11. Tolubinsky and Kostanchuk [37] predicted the bubble departure diameter by relating it to the degree of local liquid subcooling (ΔT_{sub}), which has been used with some success for a limited set of flow parameters. Another model by Kocamustafaogullari [38] suggests that a force balance between gravity ($\boldsymbol{g}$) and surface tension 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 [39] 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 [40].

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 angle was added [41] to the model and has been incorporated into the Hibiki and Ishii model [42]. 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 [43].

2.4 MIT Flow Boiling Facility

A flow boiling facility is available at MIT, where non-intrusive techniques (high-speed video and infrared thermography) are used to capture new and unique subcooled flow boiling data on multiple parameters simultaneously [44, 45]. The quantities measured at this facility include bubble departure diameter, wall superheat (local and surface-averaged), heat transfer coefficients, active nucleation site density, and bubble growth and wait time. This facility was specifically constructed to support advanced CFD model development and it provides a complete new look at the boiling phenomena. The schematic of the flow loop is illustrated in Figure 2.9. The range of potential experimental conditions is shown in Table 2.1. 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 30 mm x 10 mm (300 mm^2) and 482.6 mm in length. This results in a total entrance region length prior to the quartz cell region of 965.2 mm. The quartz cell region has a total length of 220 mm and maintains the same rectangular geometry and flow area as the entrance region and is shown in Figure 2.10. The heated area is centered in the quartz cell region and has dimensions of 10 mm x 20 mm. The heater has a thickness of 0.7 µm and is a resistively heated layer of Indium-Tin-Oxide (ITO) using graphite electrodes deposited over a sapphire substrate (1 mm thick). The ITO layer is opaque to IR in the 3-5 µm range, while the sapphire

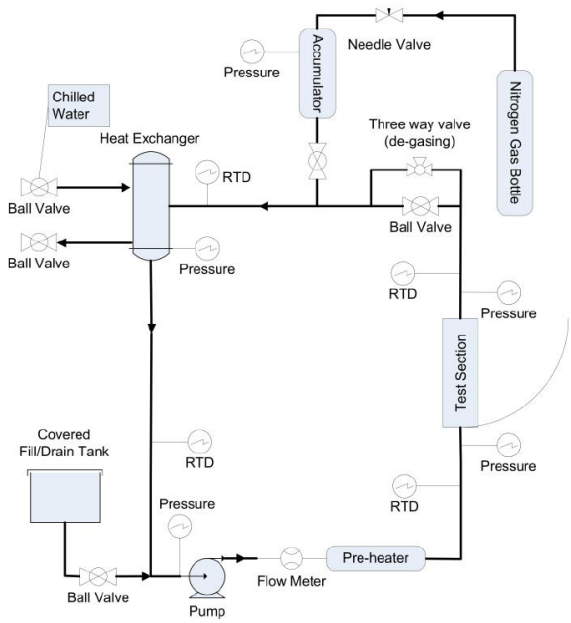


Figure 2.9. Diagram of the flow boiling loop. Courtesy of Dr. Greg DeWitt.

Table 2.1. Potential experimental conditions for the MIT Flow Boiling Facility.

Parameter	Experiment Range
Inlet water velocity	0-2 m/s
Mass flux	0-1800 kg/m^2-s
Temperature	25-250°C
Pressure	0.1-0.4 MPa
Subcooling	0-75°C
Heat flux	0-2 MW/m^2

substrate is IR transparent in this same range. Therefore, a temperature profile can be obtained normal to the heater since the ITO is the boiling surface. The static contact angle at room temperature for ITO with water is 100°. The heater is flush with the wall of the quartz cell so there is no disturbance of the flow over the heated region [45].

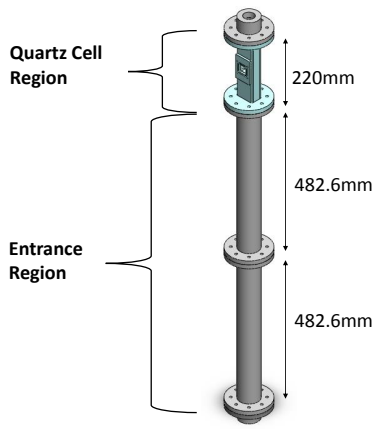


Figure 2.10. Drawing of the entrance and quartz regions provided by Bren Phillips.

Data collected at pressures of 1.05, 1.5, and 2.0 bar 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 2.11. The nucleation site density increased with increasing wall superheat and decreasing mass flux and an example is shown in Figure 2.12. Additionally, localized cooling underneath sliding bubbles was observed. This is shown in Figure 2.13 for a low heat flux (130 kW/m^2) and mass flux (200 kg/m^2-s) and 10°C subcooling [30, 44].

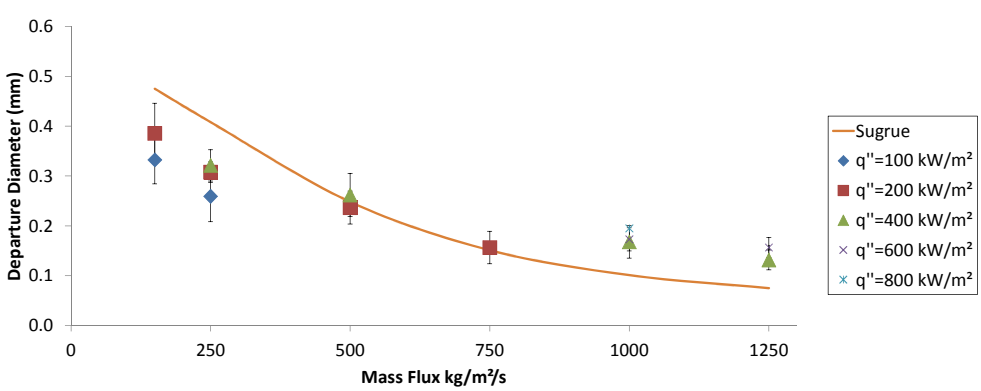


Figure 2.11. Bubble departure diameters versus mass flux for 1.05 bar and 10°C subcooling. Adapted from [44].

The velocity of a sliding bubble was approximately the same speed as the bulk flow at low mass flux and around 50-70% 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 fluxes, 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 2.14 as a function of the mass flux. The cooling under the sliding bubbles also tends to be roughly constant from 2-4°C [44].

The bubble frequency was also investigated by calculating the bubble period (τ) using the IR thermography

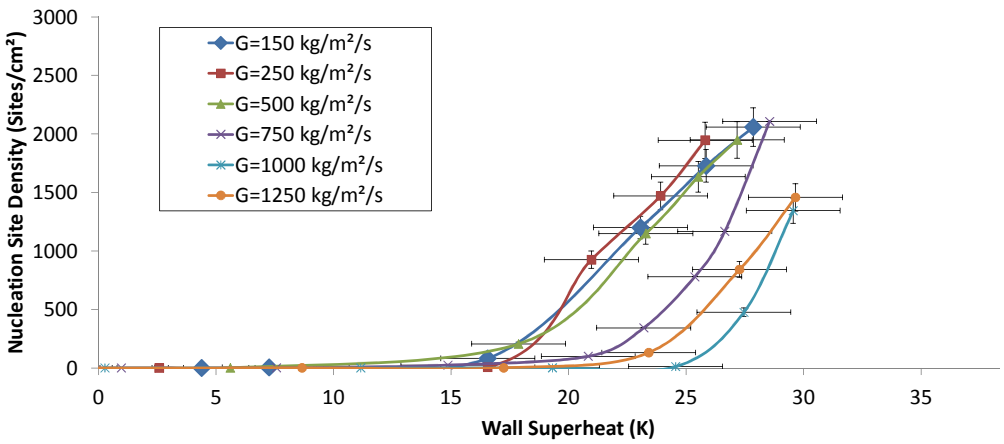


Figure 2.12. Active nucleation site densities for 1.05 bar and 10°C subcooling. Adapted from [44].

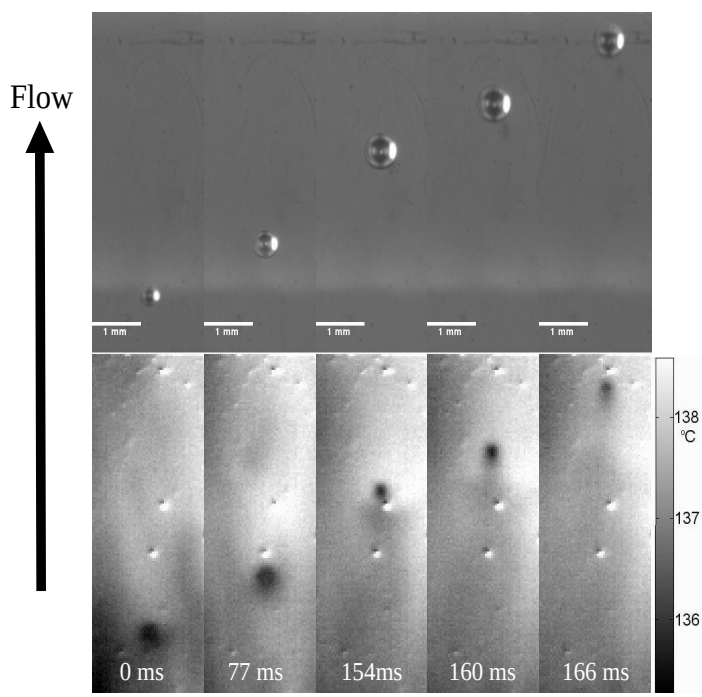


Figure 2.13. HSV (top) and IR (bottom) of a bubble from the onset of sliding (at t=0 ms) to the edge of the frame (at t=166 ms). Adapted from [44].

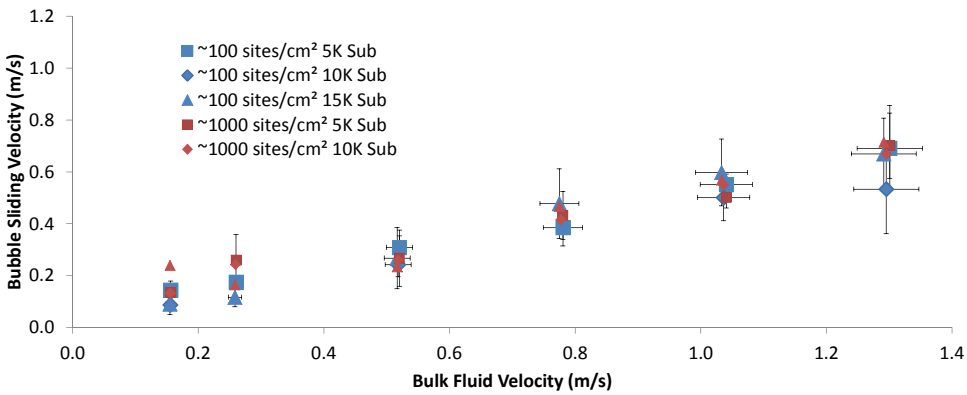


Figure 2.14. The bubble sliding velocities versus the bulk fluid velocity at 1.05 bar. Adapted from [44].

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 2.15 shows that the bubble period decreases as the wall superheat decreases. The values are compared to the bubble frequency model by Basu [40] and the experimental values consistently have a larger bubble period.

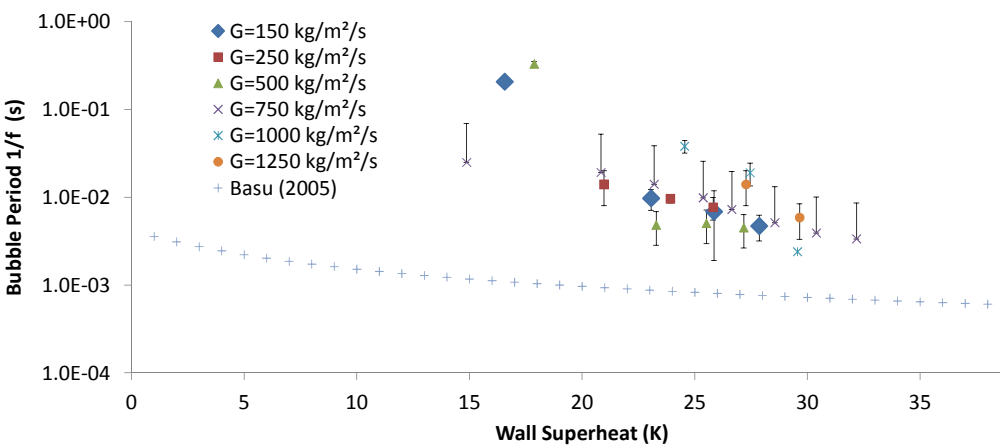


Figure 2.15. The bubble period versus wall superheat at 1.05 bar and 10°C subcooling. Adapted from [44].

2.5 MIT Flow Boiling Facility CFD Study

CFD simulations were performed in order to assess the flow conditions in the MIT Flow Boiling Facility and identify any potential anomaly which should be understood before starting the test data collection. In particular, attention is focused on the influence of both the inlet and outlet sections on the heated region flow conditions. These simulations have been performed as single-phase.

2.5.1 Geometry

The channel geometry was built inside the 3D-CAD (Computer Aided Design) parametric solid modeler of STAR-CCM+. The quartz cell region dimensions were used for the CAD with an imprinted region in the center to represent the heater. A portion of the circular outlet was included to also investigate the effect of the change from the rectangular channel geometry to the outlet circular flow area. While only a short domain is modeled in the CAD, as shown in Figure 2.16, the inlet and outlet sections are extruded during the meshing process for an optimal mesh distribution and the parameters are given in Table 2.2.

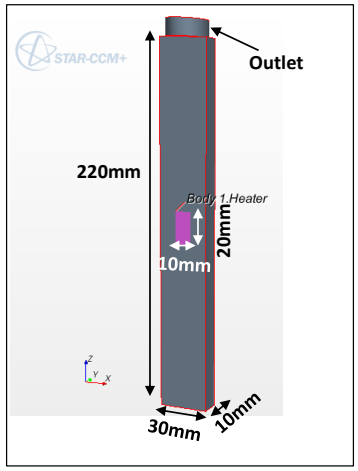


Figure 2.16. Original CAD geometry of quartz cell region.

Table 2.2. Extrusion parameters from the original CAD geometry.

	Quartz Inlet	Tube Outlet
Extrusion type	Hyperbolic tangent	Hyperbolic tangent
Number of layers	200	50
Extrusion length	965.2 mm (-z)	60 mm (+z)

The trimmed cell mesher of STAR-CCM+ was used to produce a hexa-dominant mesh, in combination with the boundary fitted prism layers in the near wall region. For mesh economy, an aspect ratio of two is used in the heater region where the cells are elongated in the flow direction. The mesh extrusion capability of STAR-CCM+ is used to model the full geometry and is shown in Figure 2.17 with the final dimensions. The parameters and values used in creating the meshed geometry are shown in Table 2.4. The mesh extrusion is completed at the inlet and outlet. The number of prism layers, prism layer thickness, and thickness of the near wall prism layer were determined by analyzing the wall y^+ values for the base case simulation. The wall y^+ values is the non-dimensional distance from the wall, and is described and defined in Section 6.1. Figure 2.18 illustrates the surface mesh around the heater while Figure 2.19 shows a cross-section of the volume mesh perpendicular to the flow at the center of the heater.

The multi-phase y^+ wall treatment is used in modeling this channel, as is characteristic of the Eulerian Multi-Phase (EMP) method, and consistent with the macroscopic representation of the boiling phenomena at the wall. The wall function approach further allows the reduction of the number of cells required to explicitly solve for velocities of the phases near the wall, but requires exact control of the thickness of the first layers. To provide for the correct cell size near the wall to utilize this model, the thickness of the near

Table 2.3. Meshing values used for building the reference base case model.

Parameter	Value
Base size	0.9 mm
Maximum cell size (relative to base)	100%
Number of prism layers	2
Prism layer thickness	0.52 mm
Thickness of near wall prism layer	0.22 mm
Surface growth rate	1.3
Template growth rate	Fast

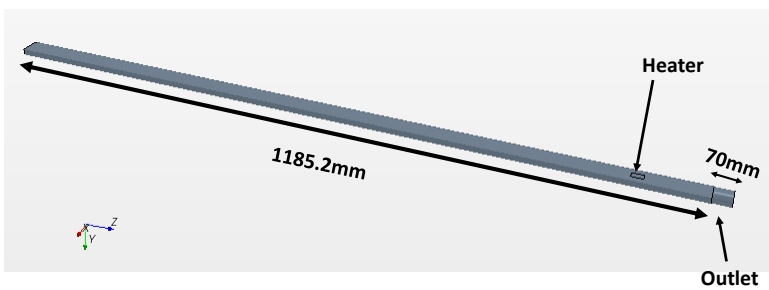


Figure 2.17. Extruded geometry to model the entire length of the channel.

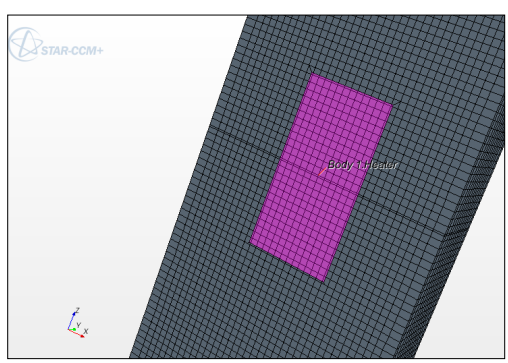


Figure 2.18. Surface mesh at the location of the heater.

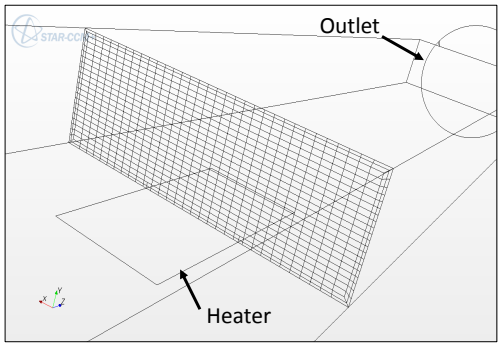


Figure 2.19. Cross-section of the volume mesh at the center of the heater perpendicular to the flow.

wall prism layer and prism layer thickness (using two prism layers) were investigated to confirm the wall y^+ of water to be in the range of 30-40. Figure 2.20 illustrates the wall y^+ values to be ~35 in this region of the channel.

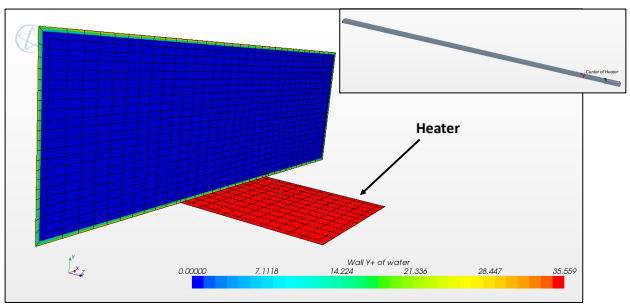


Figure 2.20. Wall y^+ values in the heater region.

2.5.2 Flow Analysis

A reference base case was chosen to implement initial conditions and investigate characteristics of the flow in single-phase flow simulations. The base case conditions are shown in Table 2.4 and were chosen based off the thermal-hydraulic conditions of the MIT flow boiling facility [45].

Table 2.4. Physics continuum initial conditions for the reference base case.

Physics Parameter	Value
Pressure	0.1 MPa
Inlet Velocity	2.0 m/s
Inlet Temperature	90°C
Turbulent Velocity Scale	2 m/s
Turbulent Length Scale	2.0 mm

Due to the rectangular configuration of the test section, anisotropy driven secondary flows are expected to develop (secondary flows of Prandtl's second kind) and must be accurately modeled in order to correctly

predict flow and temperature distributions in the test section. In order to account for secondary flows in the channel, the Improved Anisotropic Turbulence Model was adopted [46] as it has undergone extensive validation for these types of flows. Instead of the customary eddy-viscosity assumption, a cubic stress-strain relation is implemented using improved non-linear, quadratic, and cubic coefficients.

To demonstrate the importance of the anisotropic turbulence model to capture recirculation in the developed flow through the channel, the secondary flow was investigated at a location near the heater and is shown in Figure 2.21. These secondary flows have an average recirculation velocity of ∼0.02 m/s, encompass ∼1% of the inlet velocity flow of 2.0 m/s in the +z direction. It can be seen that the secondary flow scale is very low, but while they do not play an immediate action, they do have a strong influence in the overall distribution of velocity and turbulence (and consequently temperature). These results provide an insight into the importance of secondary flows that need to be included in the boiling simulations.

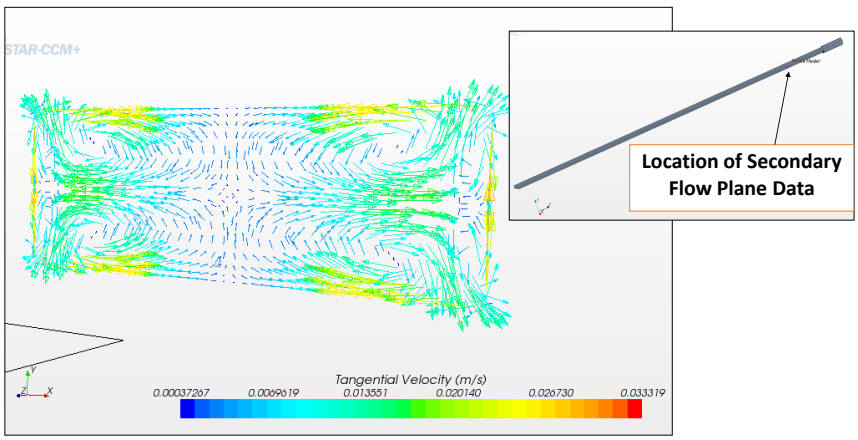


Figure 2.21. Secondary flow velocities with an initial inlet velocity of 2.0 m/s in the +z direction.

At the outlet of the channel, the 10 mm x 30 mm (300 mm^2 flow area) rectangular channel abruptly changes to a circular geometry with a diameter of 24 mm (452.4 mm^2 flow area). This circular outlet geometry was included in the CAD model to investigate the influence of this flow section and geometry change on the flow direction and velocity near the heater. The results show that the area change is sufficiently far from the heater and does not have any influence on the measurement region. As Figure 2.22 indicates, the direction and velocity of the flow near the heater does not experience any backflow due to the outlet geometry change.

A flexible tube is attached at the rectangular inlet of the entrance region that the water flows through to enter the test section. As the water flows around the flow loop though this tube and into the entrance region, it passes through a bend with an angle ranging from 0-90° depending on the orientation of the channel. To study the effect that this bend prior to the entrance region may have on the velocity profile at the heater location, an inlet velocity gradient was imposed in place of the constant 2.0 m/s (+z). The velocity gradient imposed is a function of the x-position, and is shown in (2.4). The x- and y-direction inlet velocities were maintained at 0.0 m/s as it was in the constant inlet velocity case. This gradient provides an average inlet velocity of 2.0 m/s and the fully developed velocity profile at the heater is compared with the fully-developed velocity profile for the constant inlet velocity in Figure 2.23. This figure shows that even with an extreme velocity gradient as an inlet condition, the velocity profile at the heater location is similar to the constant velocity inlet case.

$$v_z = 133.3x \tag{2.4}$$

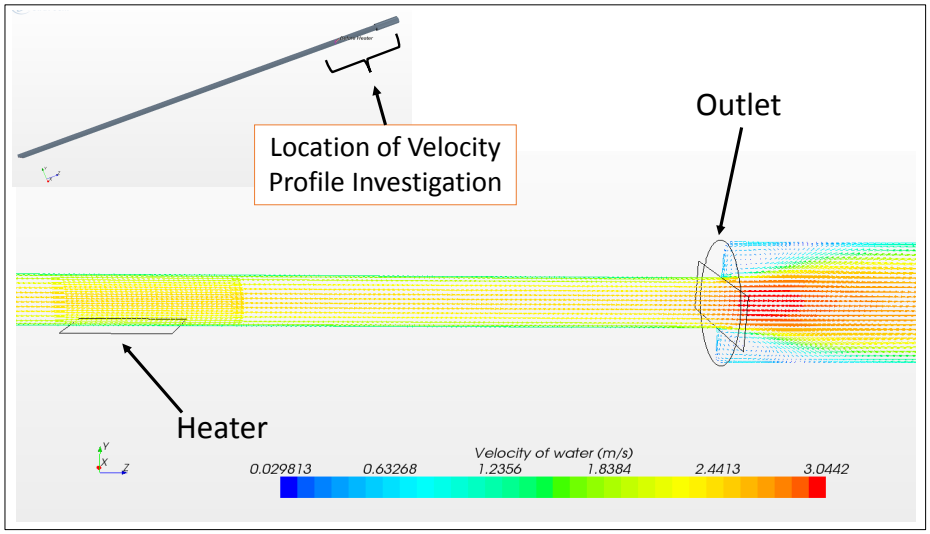


Figure 2.22. Flow direction analysis near the heater to determine the influence of the outlet geometry.

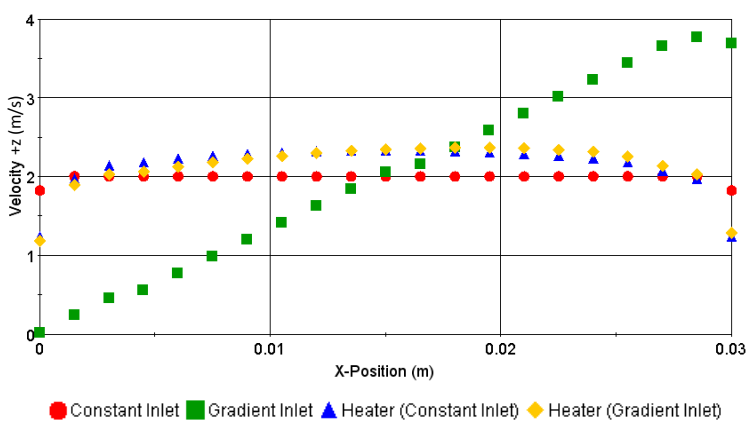


Figure 2.23. Comparison of the velocity profile at the location at the center of the heater for a constant inlet velocity and an inlet velocity that varies in the x-direction.

3 Current Wall Boiling Partitioning Approach

3.1 Eulerian Multi-Phase Framework in CFD

The testing and implementation of the wall boiling models is completed using the commercial software STAR-CCM+. The Eulerian Multi-Phase (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. Each fluid has three governing equations that must be conserved- mass, momentum, and energy [47]. The EMP model defines the influence of the phases upon eachother through interfacial terms.

In general, a multi-phase flow is any situation that involves two or more phases moving simultaneously [48]. In wall boiling, the continuous phase is the liquid water and the dispersed phase is the water vapor. The liquid and vapor phases exhibit motion among the phases in addition to heat transfer across a phase boundary. Two-fluid modeling averages the local instantaneous conservation equations over each phase. They can be averaged in time, with T as the averaging time scale, space, with V as the volume based on an averaging length scale, or some combination of these.

The use of averaging allows the solution to be tractable, but this requires that the information regarding local gradients between phases be re-supplied in the form of semi-empirical closure relationships. These closure relationships are vital for the success of the prediction of the multi-phase flow. The governing equations of mass, momentum, and energy are often formulated using the averaging over a volume V. The averaging must be done over a volume that contains a representative sample of each phase [47, 48].

Each equation of fluid motion is obtained from fundamental principles from conservation laws of physics- conservation of mass, Newton's second law for the conservation of momentum, and the first law of thermodynamics for the conservation of energy [48]. The fundamental principle that governs the conservation of mass is that the rate of increase of mass in a fluid element is equal to the net rate that mass enters the elemental volume. The local and instantaneous equation for the conservation of mass for the kth phase is given in (3.1) where $\boldsymbol{U}_k = (u_k, v_k, w_k)$ and u_k, v_k, and w_k are the velocities in the x-, y-, and z-directions respectively.

$$\frac{\partial \rho_k}{\partial t} + \frac{\partial(\rho_k U_k)}{\partial x} = \frac{\partial \rho_k}{\partial t} + \nabla\cdot(\rho_k \boldsymbol{U}_k) = 0 \tag{3.1}$$

In two-phase flow, the local volume fraction (α_k) is used to solve each governing equation for each phase. The addition of the local volume fractions for each phase must be equal to one. This allows the instantaneous equation for the Reynolds-averaged conservation of mass to be as shown in (3.2). The right-hand side of the equation is known as the interfacial mass source (Γ_k), where $\dot{m}_{lk}$ and $\dot{m}_{kl}$ capture the mass transfer from the lth phase to the kth phase and the kth phase to the lth phase respectively [48].

$$\frac{\partial(\alpha_k \rho_k)}{\partial t} + \nabla\cdot(\alpha_k \rho_k \boldsymbol{U}_k) = \sum_{l=1}^{2}(\dot{m}_{lk} - \dot{m}_{kl}) \tag{3.2}$$

Using the definition of Newton's second law that the rate of increase of momentum of a fluid element is equal to the sum of the forces acting on the fluid element, the conservation of momentum equation is derived. The rate of increase of the momentum of the fluid element is given in (3.3) while the two forces acting on the fluid element are called the surface and body forces. These two forces are implemented using additional source terms to the momentum equation.

$$\rho_k \frac{Du_k}{Dt} dx\, dy\, dz \tag{3.3}$$

The local instantaneous equation for the conservation of momentum is given in (3.4), where p_k is the kth phase pressure. The stress tensor $\boldsymbol{\tau}_k$ is made up of the components $(\tau_{xx}^k, \tau_{xy}^k, ..., \tau_{yz}^k, \tau_{zz}^k)$. The body forces $\boldsymbol{F}^{k,bodyforces}$ vector is made of the body forces in the x-, y-, and z-directions [48]. An example of a common body force included is the body force due to gravity.

$$\rho_k \frac{\partial \boldsymbol{U}_k}{\partial t} + \rho_k \boldsymbol{U}_k \cdot \nabla \boldsymbol{U}_k = -\nabla p_k + \nabla \cdot \boldsymbol{\tau}_k + \sum \boldsymbol{F}^{k,bodyforces} \tag{3.4}$$

The equation for the Reynolds-averaged conservation of momentum for each phase is shown in (3.5). The last four terms on the right-hand side of the equation are known as the interfacial momentum sources, $\boldsymbol{\Omega}_k$, where the interfacial force density (F_D^k) is shown as two terms- the drag force $(\boldsymbol{F}_D^{k,drag})$ and the non-drag interfacial forces $(\boldsymbol{F}_D^{k,non-drag})$. The dynamic viscosity of the kth phase is given as μ_k and $\boldsymbol{U}_l$ is the velocity vector for the lth phase. $\boldsymbol{\tau}_{k''}$ relates the Reynolds stress for the different phases to the averaged velocity as is shown in (3.6). Often, it is assumed that both phases have the same pressure so that $p_k = p$ [48].

$$\begin{aligned}\frac{\partial(\alpha_k \rho_k \boldsymbol{U}_k)}{\partial t} + \nabla \cdot (\alpha_k \rho_k \boldsymbol{U}_k \otimes \boldsymbol{U}_k) = -\alpha_k \nabla p_k + (\nabla \cdot \alpha_k [\mu_k (\nabla \boldsymbol{U}_k + (\nabla \boldsymbol{U}_k)^T) - \frac{2}{3}\mu_k \nabla \cdot \boldsymbol{U}_k \delta]) - \nabla \cdot (\alpha_k \boldsymbol{\tau}_{k''}) \\ + \alpha_k \rho_k \boldsymbol{g} + \sum_{l=1}^{2} (\dot{m}_{lk} \boldsymbol{U}_l - \dot{m}_{kl} \boldsymbol{U}_k) + (p_{int}^k - p) \nabla \alpha_k + \boldsymbol{F}_D^{k,drag} + \boldsymbol{F}_D^{k,non-drag}\end{aligned} \tag{3.5}$$

$$-\boldsymbol{\tau}_{k''} = \mu_{k,T} (\nabla \boldsymbol{U}_k + (\nabla \boldsymbol{U}_k)^T) - \frac{2}{3}\mu_{k,T} \nabla \cdot \boldsymbol{U}_k \delta - \frac{2}{3}\rho_k k_k \delta \tag{3.6}$$

The turbulent viscosity for the kth phase is represented by $\mu_{k,T}$ and is shown in (3.7) for a characteristic length scale of $(k_k)^{3/2}/\varepsilon_k$. The turbulent kinetic energy for the kth phase is given as k_k and the dissipation of turbulent energy is represented as ε_k. The constant C_μ is referred to as the turbulent viscosity scaling and is set equal to 0.09 [49].

$$\mu_{k,T} = C_\mu \rho_k \frac{(k_k)^2}{\varepsilon_k} \tag{3.7}$$

The conservation of momentum governing equation (3.5) captures the Navier-Stokes equations that govern the motion of the phases [48]. The typical formulation of the Navier-Stokes equations for a single-phase in conservation form are given in (3.8), (3.9), and (3.10) for the x-, y-, and z-directions respectively. The body forces in the x-, y-, and z-direction are given by f_x, f_y, and f_z respectively [47].

$$\frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u \boldsymbol{U}) = -\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + \rho f_x \tag{3.8}$$

$$\frac{\partial(\rho v)}{\partial t} + \nabla \cdot (\rho v \boldsymbol{U}) = -\frac{\partial p}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + \rho f_y \tag{3.9}$$

$$\frac{\partial(\rho w)}{\partial t} + \nabla \cdot (\rho w \boldsymbol{U}) = -\frac{\partial p}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} + \rho f_z \tag{3.10}$$

Employing the first law of thermodynamics, the equation for energy is first derived assuming that the rate of increase of energy in a fluid element is equal to the net rate of heat added to the element and the net rate of work done on the fluid element. The total enthalpy equation, which is the most convenient form and is often applied for multi-phase flow studies, is shown in (3.11). Here, Φ_H^k describes the effects due to the viscous extra stresses in the conservation equation and is defined as $\nabla \cdot (\boldsymbol{U}_k \cdot \boldsymbol{\tau}_k)$. H_k is the total enthalpy of the fluid,

and it contains both the specific internal energy (e_k) and the kinetic energy $\left(\frac{1}{2}[u_k u_k + v_k v_k + w_k w_k]\right)$ of the fluid and is shown in (3.12). The heat addition in each direction is represented by the vector $\boldsymbol{q}$ [48].

$$\rho_k \frac{\partial H_k}{\partial t} + \rho_k \boldsymbol{U}_k \cdot \nabla H_k = -\frac{\partial p_k}{\partial t} - \nabla \cdot \boldsymbol{q}_k + \sum \boldsymbol{F}^{k,bodyforces} \cdot \boldsymbol{U}_k + \Phi_H^k \tag{3.11}$$

$$H_k = e_k + \frac{p_k}{\rho_k} + \frac{1}{2}(u_k u_k + v_k v_k + w_k w_k) \tag{3.12}$$

The final form of the Reynolds-averaged energy equation for the kth phase is shown in (3.13). The interfacial heat source (Q_H^{int}) is shown in (3.14) where C_{kl} is the inter-phase heat-transfer term, T_l is the temperature of the lth phase, and T_k is the temperature of the kth phase. The thermal conductivity of the kth phase is represented as λ_k and $Pr_{k,T}$ is the turbulent Prandtl number. The final two terms on the right-hand side of the (3.13) represent the interfacial energy sources, Π_H^k.

$$\begin{aligned}\frac{\partial(\alpha_k \rho_k H_k)}{\partial t} + \nabla \cdot (\alpha_k \rho_k \boldsymbol{U}_k H_k) &= \nabla \cdot (\alpha_k \lambda_k \nabla T_k) - \nabla \cdot \left(\alpha_k \frac{\mu_{k,T}}{Pr_{k,T}} \nabla H_k\right) \\ &+ \sum_{l=1}^{2} (\dot{m}_{lk} H_l - \dot{m}_{kl} H_k) + Q_H^{int}\end{aligned} \tag{3.13}$$

$$Q_H^{int} = \sum_{l=1}^{2} C_{kl}(T_l - T_k) \tag{3.14}$$

3.2 Modeling of Turbulent Flows

Typically the flows encountered in the practice of engineering are turbulent. Turbulent flows are characterized by high vorticity and the vorticity increases the rate of mixing through turbulent diffusion. The increased mixing is desirable for high heat transfer. Turbulence reduces the velocity gradients in the mean bulk flow due to the action of viscosity, which reduces the kinetic energy of the flow. This lost energy is converted to internal energy of the fluid [50].

There are multiple approaches for the prediction of turbulent flow and the most common approaches are listed below [50].

- The use of correlations for prediction of the friction factor and heat transfer as a function of the Reynolds and Prandtl numbers does not require a computer, but it is limited to simple types of flows.
- The Reynolds-averaged Navier Stokes (RANS) method averages the equations of motion. If the flow is statistically steady, each variable is written as the sum of the time-averaged variable plus a fluctuation about that value. The averaging time interval (T) must be large compared to the fluctuation time scale. Therefore, when T is large enough, the averaged value is time-independent. These equations were detailed in Section 3.1. For unsteady flow (URANS), ensemble averaging is used in place of time averaging. Both RANS and URANS approaches require the use of turbulence models to close the set of equations [50].
- The large eddy simulation (LES) approach resolves the largest scale motions, but models the small scale motions of the flow.
- The direct numerical simulation (DNS) approach solves the Navier-Stokes equations for all scales of motion. This is the most accurate approach for simulating turbulence because no averaging or approximating is needed to solve the Navier-Stokes equations, but it is computationally expensive because the size of the grid can be no larger than a viscously determined scale (Kolmogoroff scale).

3.3 Current Partitioned Heat Flux Boiling Model

There are various approaches for modeling two-phase flow using CFD, and the baseline boiling method uses the EMP framework using the RANS turbulence modeling approach [51]. The wall boiling representation in this framework is nevertheless still relying on the classic heat partitioning concept introduced by Judd and Hwang (1976) [52] and adapted by Kurul and Podowski (1990) [53] and is shown in (3.15) and illustrated in Figure 3.1. In the simulation, the total heat flux (q''_{tot}) is computed as the sum of the partitioned components. The convection term (q''_{fc}) describes the removal of heat by single-phase turbulent convection and is shown in (3.16), where the heat transfer coefficient is represented by h_{fc}. The quenching term (q''_q) describes the enhancement of heat transfer due to the replacement of a departing bubble by an influx of cooler liquid and is shown in (3.17).

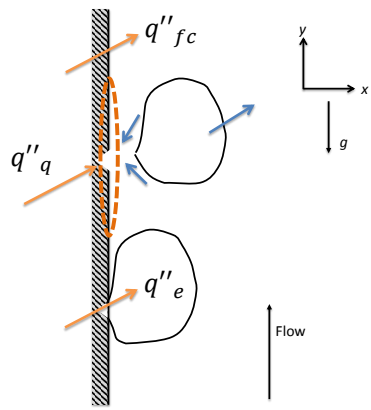


Figure 3.1. Depiction of the original heat flux partitioning for subcooled flow boiling.

$$q''_{tot} = q''_{fc} + q''_q + q''_e \tag{3.15}$$

$$q''_{fc} = h_{fc}(\Delta T_{sup} + \Delta T_{sub}) \tag{3.16}$$

The Del Valle and Kenning model [32] is used to determine the quenching heat transfer coefficient and is shown in (3.18). 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 to account for the scaling from the nucleation site density to the wall area fraction that the quenching component influences. This is given in (3.19) where F_A represents the additional area that is under the influence of quenching and is set equal to 2.0. This is the common value used through implementation experience and from the results of a study completed on the Bartolomei and Chanturiya test case [54, 55]. The liquid density, liquid specific heat, and liquid thermal conductivity are given by ρ_l, c_{pl}, and k_l respectively.

$$q''_q = h_{quench}(\Delta T_{sup} + \Delta T_{sub}) \tag{3.17}$$

$$h_{quench} = 2K_{quench} f \sqrt{\frac{\rho_l c_{pl} k_l t_w}{\pi}} \tag{3.18}$$

$$K_{quench} = F_A \frac{\pi D_d^2}{4} N'' \tag{3.19}$$

The evaporation heat flux (q_e'') captures the heat required to form the vapor phase and is shown in (3.20), where ρ_v is the vapor density and h_{fg} is the latent heat of evaporation. The bubble departure diameter, active nucleation site density, and frequency for bubble departure are determined using correlations.

$$q_e'' = \rho_v h_{fg} \frac{\pi}{6} D_d^3 N'' f \tag{3.20}$$

The bulk boiling/condensation is driven by the heat transfer between phases. The heat transfer coefficient for the gas side uses a constant value while the liquid side typically adopts the Ranz-Marshall correlation [56]. This is shown in (3.21) for calculation of the Nusselt number (Nu) where $\mathrm{Re_b}$ (3.22) is the dispersed phase (vapor) Reynolds number, often referred to as the bubble Reynolds number, and $\mathrm{Pr_l}$ (3.24) is the continuous phase (liquid) Prandtl number. The liquid dynamic viscosity is given by μ_l. The velocity at the bubble center is given by $U_{b,rel}$ and is shown in (3.23) where $U_{b,z}$ and $U_{b,y}$ are the instantaneous velocities at the bubble center in the axial (z) and normal (y) directions respectively. U_l is the bulk liquid velocity in the axial direction. The velocity in the x-direction is assumed to be negligent.

$$\mathrm{Nu} = 2 + 0.6{\mathrm{Re_b}}^{0.5}{\mathrm{Pr_l}}^{0.3} \tag{3.21}$$

$$\mathrm{Re_b} = \frac{\rho_l U_{b,rel} D_d}{\mu_l} \tag{3.22}$$

$$U_{b,rel} = \sqrt{(U_{b,z} - U_l)^2 + (U_{b,y})^2} \tag{3.23}$$

$$\mathrm{Pr_l} = \frac{\mu_l c_{p,l}}{k_l} \tag{3.24}$$

The Chen-Mayinger correlation [57] shown in (3.26) is often used in place of the Ranz-Marshall correlation in the standard cases because it was developed using a larger range of Prandtl (2< $\mathrm{Pr_l}$ <15) and Jakob (1< Ja <120) numbers and is more applicable for higher pressure cases. Ja is the Jakob number and is shown in (3.25). Chen and Mayinger studied the heat transfer at the vapor-liquid interface for both attached and departing bubbles using ethanol, propanol, R113, and water as the working fluids. The Nusselt number is used to calculate the heat transfer coefficient (h_l) as is shown in (3.27) [58].

$$\mathrm{Ja} = \frac{\rho_l c_{pl} \Delta T_{sup}}{\rho_v h_{fg}} \tag{3.25}$$

$$\mathrm{Nu} = 0.185{\mathrm{Re_b}}^{0.7}{\mathrm{Pr_l}}^{0.5} \tag{3.26}$$

$$h_l = \frac{\mathrm{Nu} k_l}{D_d} \tag{3.27}$$

Another condensation model, developed by Warrier et al. [58], was studied for low pressure subcooled flow boiling regimes using water as the working fluid. This correlation is applicable over the ranges of 1.8< $\mathrm{Pr_l}$ <2.9 and 12< Ja <100 and is shown in (3.28). The Fourier number (Fo) is shown in (3.29), where D_{bo} is the initial bubble lift-off diameter as a function of time. Essentially, the first part of the Nusselt number equation ($0.6{\mathrm{Re_b}}^{0.5}{\mathrm{Pr_l}}^{1/3}$) accounts for the forced convection around the bubble, while $\left[1 - 1.20\mathrm{Ja}^{9/10}\mathrm{Fo}^{2/3}\right]$ accounts for the shrinking of the bubble as condensation occurs. The condensation model by Park et al. [59] was developed using annular condensation of the refrigerant FC-72. Since the Chen-Mayinger correlation is applicable over the widest range of conditions, and the studies are completed at both low and high pressure, it was selected as the condensation model.

$$\mathrm{Nu} = 0.6\mathrm{Re_b}^{0.5}\mathrm{Pr_l}^{1/3}\left[1 - 1.20\mathrm{Ja}^{9/10}\mathrm{Fo}^{2/3}\right] \tag{3.28}$$

$$\mathrm{Fo} = \frac{\eta_l t}{D_{bo}^2} \tag{3.29}$$

The interface between the two phases is assumed to be at saturation temperature. The interaction length scale is assumed constant at 1.0 mm. A population balance model can also be implemented to better predict bubble size, void distribution, and liquid velocity profiles in a heated channel [60].

The standard bubble departure frequency model implemented is that by Cole (1960) [39] and is shown in (3.30). This departure model is a ratio of the rising bubble velocity over the bubble departure diameter, where g is the gravity force. The typical bubble departure diameter model used is the Tolubinsky-Kostanchuk [37], and is shown in (3.31). It correlates the bubble departure diameter with the liquid subcooling, and uses the constants $D_0 = 0.6$ mm and $\Delta T_0 = 45$ ° C.

$$f = \sqrt{\frac{4}{3}\frac{g(\rho_l - \rho_v)}{D_d \rho_l}} \tag{3.30}$$

$$D_d = D_0 \exp\left(-\frac{\Delta T_{sub}}{\Delta T_0}\right) \tag{3.31}$$

The Lemmert-Chawla active nucleation site density correlation [5] is the most commonly adopted model and the original formula is shown in (3.32). The default values for m and p are 185.0 sites/m^2 and 1.805 respectively. Many CFD software implementations of the Lemmert and Chawla model use a modified form that allows the exponent to increase with wall superheat [55]. This modified version is shown in (3.33) where A and B are calibration constants. When employed with the Tolubinsky-Kostanchuk bubble departure diameter model, A, B, and N_0'' are equal to 1.805, 0, and $(185.0)^{1.805}$ respectively. If the Lemmert-Chawla nucleation site density model is employed with the Kocamustafaogullari bubble departure diameter model [38], the model employs different values of A, B, and N_0'' by calibrating it against the Hibiki-Ishii nucleation site density model [42]. In all cases, ΔT_0 is set equal to 1K.

$$N'' = (m\Delta T_{sup})^p \tag{3.32}$$

$$\frac{N''}{N_0''} = \left(\frac{\Delta T_{sup}}{\Delta T_0}\right)^{A+B(\Delta T_{sup}/\Delta T_0)} \tag{3.33}$$

The Hibiki-Ishii active nucleation site density model provides good predictions over a very wide range of flow conditions and can also account for different boiling surfaces by including a dependence on the contact angle [42]. Studies have been completed to illustrate that the large spread of data on the active nucleation site density can be captured in the nucleation site density models by the inclusion of a dependence on the contact angle of the fluid on the heated material [41]. The Hibiki-Ishii model is also dependent on the system pressure, which is known to affect the nucleation site density. This model is shown in (3.34) where $\overline{N''}$ = 4.72 x 10^5 sites/m^2, μ is a fixed reference value for the contact angle set to 0.722 rad, and λ' is the cavity length scale and is set equal to 2.50 x 10^{-6} m. The minimum cavity radius (R_c) is given in (3.35) where R_g is the gas constant based on the molecular weight of the fluid. The density function is given in (3.36) where ρ^+ is shown in (3.37). Many of the systems of interest are at pressures above atmospheric, so the Hibiki-Ishii active nucleation site density model is used in this study.

$$N'' = \overline{N''}\left(1 - \exp\left(-\frac{\theta^2}{8\mu^2}\right)\right)\left[\exp\left(f(\rho^+)\frac{\lambda'}{R_c}\right) - 1\right] \tag{3.34}$$

$$R_c = \frac{2\sigma\left\{1 + (\rho_v/\rho_l)\right\}/p}{\exp\left\{h_{fg}(\Delta T_{sup})/(R_g T_w T_{sat})\right\} - 1} \tag{3.35}$$

$$f(\rho^+) = -0.01064 + 0.48246\rho^+ - 0.22712{\rho^+}^2 + 0.05468{\rho^+}^3 \tag{3.36}$$

$$\rho^+ = \log\left(\frac{\rho_l - \rho_v}{\rho_v}\right) \tag{3.37}$$

The drag force calculates the force on a dispersed phase particle, so in the simulation that involves boiling, the force acts on the bubble. The basic formulation for the drag force acting on phase i due to the drag of phase j is shown in (3.38), where $\boldsymbol{v}_j$ is the velocity vector of phase j and $\boldsymbol{v}_i$ is the velocity vector of phase i. The linearized drag coefficient (A_{ij}^D) is calculated as shown in (3.39), where the drag coefficient is C_D. The factor a_{ij} is the interfacial area density and for an equivalent spherical particle, the projected area is then $a_{ij}/4$ [55].

$$\boldsymbol{F}_{ij}^D = A_{ij}^D\left(\boldsymbol{v}_j - \boldsymbol{v}_i\right) \tag{3.38}$$

$$A_{ij}^D = C_D\frac{1}{2}\rho_j\left|\boldsymbol{v}_j - \boldsymbol{v}_i\right|\left(\frac{a_{ij}}{4}\right) \tag{3.39}$$

The basic Schiller-Naumann drag formulation is used when the test involves solid spherical particles, droplets, or small-diameter bubbles [61] and is shown in (3.40). The Tomiyama drag formulation is built specifically for systems involving bubbles and is applicable over a wider range of conditions while accounting for varying degrees of contamination of the system [62, 63]. The Tomiyama formulation is typically used for flow boiling [64]. The contaminated drag coefficient correlation is employed since most systems do not use liquid of high purity and is shown in (3.41) where Eo is the Eötvös number shown in (3.42).

$$C_D = \begin{cases} \frac{24}{\mathrm{Re_b}}\left(1 + 0.15{\mathrm{Re_b}}^{0.687}\right) & \text{for } 0 < \mathrm{Re_b} \leq 1000; \\ 0.44 & \text{for } \mathrm{Re_b} > 1000. \end{cases} \tag{3.40}$$

$$C_D = \max\left[\left(\frac{24}{\mathrm{Re_b}}(1 + 0.15{\mathrm{Re_b}}^{0.687})\right), \frac{8\mathrm{Eo}}{3(\mathrm{Eo} + 4)}\right] \tag{3.41}$$

$$\mathrm{Eo} = \frac{(\rho_l - \rho_v)gD_d^2}{\sigma} \tag{3.42}$$

The modeling of turbulence in two-phase flow is generally more complicated as compared to single-phase flow. The mechanical energy is partitioned into rotational and translational energies due to the inertia and friction forces that are caused by the increase in vorticity and strain. Additionally, the presence of the dispersed phase in the continuous phase can affect the turbulence [48].

The standard k-ε turbulence model developed by Launder and Spalding [49] is the most commonly used turbulence model in single-phase turbulent fluid problems. The single-phase turbulence model is implemented as a two-fluid model [65] by generalizing the equations and accounting for the fraction of time a

phase occupies a given point in space using the local volume fraction of the phase α_k. This modifies the transport equations for the continuous phase and they are shown in (3.43) and (3.44).

$$\frac{\partial}{\partial t}(\alpha_k \rho_k k_k) + \nabla\cdot(\alpha_k \rho_k \boldsymbol{U}_k k_k) = \nabla\cdot\left(\alpha_k \frac{(\mu_{k,T}+\mu_k)}{\sigma_k}\nabla k_k\right) + \alpha_k(P_k + G_k - \rho_k\varepsilon_k) + S_{k2}^{int} \tag{3.43}$$

$$\begin{aligned}\frac{\partial}{\partial t}(\alpha_k \rho_k \varepsilon_k) + \nabla\cdot(\alpha_k \rho_k \boldsymbol{U}_k \varepsilon_k) &= \nabla\cdot\left(\alpha^k \frac{(\mu_{k,T}+\mu_k)}{\sigma_\varepsilon}\nabla \varepsilon_k\right) \\ &+ \alpha_k \frac{\varepsilon_k}{k_k}(C_{\varepsilon 1}P_k + \|G_k\| - C_{\varepsilon 2}\rho_k\varepsilon_k) + S_{\varepsilon 2}^{int}\end{aligned} \tag{3.44}$$

Table 3.1. Constants used in the standard k-ε model.

Symbol	Value
σ_{ρ_k}	1.0
σ_k	1.0
σ_ε	1.3
$C_{\varepsilon 1}$	1.44
$C_{\varepsilon 2}$	1.92

The additional source or sink terms (S_{k2}^{int} and $S_{\varepsilon 2}^{int}$) capture the production and dissipation of turbulence from the interactions between the two phases. The shear production term P_k is given in (3.45). The production due to gravity (G_k) is shown in (3.46) and is valid for weakly compressible flows. The constants for σ_{ρ_k}, σ_k, σ_ε, $C_{\varepsilon 1}$, and $C_{\varepsilon 2}$ are shown in Table 3.1 [48].

$$P_k = \mu_{k,T}\nabla \boldsymbol{U}_k \cdot (\nabla \boldsymbol{U}_k + (\nabla \boldsymbol{U}_k)^T) - \frac{2}{3}\nabla\cdot \boldsymbol{U}_k(\mu_{k,T}\nabla\cdot \boldsymbol{U}_k - \rho_k \boldsymbol{U}_k) \tag{3.45}$$

$$G_k = -\frac{\mu_{k,T}}{\rho_k \sigma_{\rho_k}}\boldsymbol{g}\cdot\nabla\rho_k \tag{3.46}$$

In order to account for secondary flows that may occur in simulations, the Improved Anisotropic Turbulence Model was adopted [46], as it has undergone extensive validation for geometries that have anisotropy driven secondary flows. These flows must be accurately modeled in order to correctly predict flow and temperature distributions in the simulation. Instead of the customary eddy-viscosity assumption, a cubic stress-strain relation is implemented and non-linear production terms are included in the turbulent production equation as shown in (3.47). The non-linear production term (G_k^{NL}) is calculated as shown in (3.48) for the cubic model, and (3.49) for the quadratic model [55].

$$\begin{aligned}\frac{\partial}{\partial t}(\alpha_k \rho_k \varepsilon_k) + \nabla\cdot(\alpha_k \rho_k \boldsymbol{U}_k \varepsilon_k) &= \nabla\cdot\left(\alpha^k \frac{(\mu_{k,T}+\mu_k)}{\sigma_\varepsilon}\nabla \varepsilon_k\right) \\ &+ \alpha_k \frac{\varepsilon_k}{k_k}\left(C_{\varepsilon 1}\left(P_k + \|G_k\| + G_k^{NL}\right) - C_{\varepsilon 2}\rho_k\varepsilon_k\right) + S_{\varepsilon 2}^{int}\end{aligned} \tag{3.47}$$

$$G_k^{NL} = \nabla \boldsymbol{U}_k : \left(\boldsymbol{T}_{t,cubic} - \boldsymbol{T}_{t,lin}\right) \tag{3.48}$$

$$G_k^{NL} = \nabla \boldsymbol{U}_k : \left(\boldsymbol{T}_{t,quad} - \boldsymbol{T}_{t,lin}\right) \tag{3.49}$$

The calculations for $\boldsymbol{T}_{t,lin}$, $\boldsymbol{T}_{t,quad}$, and $\boldsymbol{T}_{t,cubic}$ are shown in (3.50), (3.51), and (3.52) respectively, where $\boldsymbol{S}$ is the strain rate tensor given by (3.53) and $\boldsymbol{W}$ is the vorticity tensor as defined in (3.54).

$$\boldsymbol{T}_{t,lin} = 2\mu_{k,T}\boldsymbol{S} - \frac{2}{3}\left(\mu_{k,T}\nabla\cdot\boldsymbol{U}_k + \rho k\right) \tag{3.50}$$

$$\begin{aligned}\boldsymbol{T}_{t,quad} = \boldsymbol{T}_{t,lin} &- 4C_1\mu_{k,T}\frac{k}{\varepsilon}\left[\boldsymbol{S}\cdot\boldsymbol{S} - \frac{1}{3}\left(\boldsymbol{S}:\boldsymbol{S}\right)\right] - 4C_2\mu_{k,T}\frac{k}{\varepsilon}\left(\boldsymbol{W}\cdot\boldsymbol{S} + \boldsymbol{S}\cdot\boldsymbol{W}^T\right)\\ &- 4C_3\mu_{k,T}\frac{k}{\varepsilon}\left[\boldsymbol{W}\cdot\boldsymbol{W}^T - \frac{1}{3}\left(\boldsymbol{W}:\boldsymbol{W}^T\right)\right]\end{aligned} \tag{3.51}$$

$$\boldsymbol{T}_{t,cubic} = \boldsymbol{T}_{t,quad} - 8C_4\mu - k,T\frac{k^2}{\varepsilon^2}\left[\left(\boldsymbol{S}\cdot\boldsymbol{S}\right)\cdot\boldsymbol{W} + \boldsymbol{W}^T\cdot\left(\boldsymbol{S}\cdot\boldsymbol{S}\right)\right] - 8C_5\mu_{k,T}\frac{k^2}{\varepsilon^2}\left(\boldsymbol{S}:\boldsymbol{S} - \boldsymbol{W}:\boldsymbol{W}^T\right)\boldsymbol{S} \tag{3.52}$$

$$\boldsymbol{S} = \frac{1}{2}\left(\nabla\boldsymbol{U}_k + \nabla\boldsymbol{U}_k^T\right) \tag{3.53}$$

$$\boldsymbol{W} = \frac{1}{2}\left(\nabla\boldsymbol{U}_k - \nabla\boldsymbol{U}_k^T\right) \tag{3.54}$$

Instead of a constant turbulent viscosity scaling coefficient, C_μ is calculated using the non-linear coefficients as shown in (3.55) where $S = \frac{k}{\varepsilon}\sqrt{2\boldsymbol{S}:\boldsymbol{S}}$ and $W = \frac{k}{\varepsilon}\sqrt{2\boldsymbol{W}:\boldsymbol{W}}$. The quadratic and cubic coefficients are used to calculate the remaining coefficients C_1, C_2, C_3, C_4, and C_5 as shown in (3.56). The improved non-linear, quadratic, and cubic coefficients are shown in Tables 3.2, 3.3, and 3.4 respectively.

$$C_\mu = \frac{C_{a0}}{C_{a1} + C_{a2}S + C_{a3}W} \tag{3.55}$$

$$\begin{aligned}C_1 &= \frac{C_{nl1}}{\left(C_{nl6} + C_{nl7}S^3\right)C_\mu}\\ C_2 &= \frac{C_{nl2}}{\left(C_{nl6} + C_{nl7}S^3\right)C_\mu}\\ C_3 &= \frac{C_{nl3}}{\left(C_{nl6} + C_{nl7}S^3\right)C_\mu}\\ C_4 &= C_{nl4}C_\mu^2\\ C_5 &= C_{nl5}C_\mu^2\end{aligned} \tag{3.56}$$

The virtual mass force models the interphase momentum transfer to account for additional resistance that occurs when a particle undergoes acceleration. This is done by using inviscid flow theory to represent this phenomenon as a "virtual mass" [66]. It is helpful to use in non-accelerating flows as well because it makes the simulation less sensitive to momentum or relaxation factors [55]. The two-phase formulation uses the formulation by Auton et. al. [67] and is shown in (3.57) for the virtual mass force acting on phase i due to

Table 3.2. Non-linear coefficients.

Coefficient	Value
Ca0	0.667
Ca1	3.9
Ca2	1.0
Ca3	0.0

Table 3.3. Quadratic coefficients.

Coefficient	Value
Cnl1	0.8
Cnl2	11.0
Cnl3	4.5
Cnl6	1000
Cnl7	1.0

Table 3.4. Cubic coefficients.

Coefficient	Value
Cnl4	-5.0
Cnl5	-4.5

acceleration relative to phase j. The acceleration term $(a_j - a_i)$ is given by the rate of change of the velocity of constant mass particles. C^{VM} is the virtual mass coefficient that is set equal to 0.5.

$$F_{ij}^{VM} = C^{VM} \rho_l \alpha_v [a_j - a_i] \tag{3.57}$$

The Troshko and Hassan Particle induced turbulence model [68] is used because it captures the bubble induced turbulence effects. It accounts for the effects of bubbles by adding turbulent kinetic energy and dissipation source terms to the liquid turbulence and was developed using the standard k-ε turbulence model. The turbulent kinetic energy source term (S_{k_l}) is an interfacial term that assumes low-bubble inertia which is valid for most bubbly flows. It correlates the velocity fluctuation at the bubble-liquid interface and interfacial force density, which is equal to the work of interfacial force density per unit time. It accounts for the turbulence addition caused by bubble induced mixing and is shown in (3.58). $\boldsymbol{U}_r$ is the slip velocity and is shown in (3.59) where $\boldsymbol{U}_l$ and $\boldsymbol{U}_v$ are the liquid and vapor velocity vectors respectively.

$$S_{k_l} = \frac{3}{4} \frac{C_D}{D_d} \alpha_v \rho_l \left| \boldsymbol{U}_r \right|^3 \tag{3.58}$$

$$\boldsymbol{U}_r = \boldsymbol{U}_v - \boldsymbol{U}_l \tag{3.59}$$

The interfacial dissipation term was modeled by assuming it is proportional to the bubble-induced pseudo-turbulence destruction characteristic frequency multiplied by the bubble-induced turbulent energy production term and is shown in (3.60) where it shows that it decays over a characteristic time, t_b. C_3 is a constant for incompressible bubbly flows and is equal to 0.45. This produced a new bubble pseudoturbulence dissipation frequency that states that the characteristic time scale of the bubble pseudoturbulence destruction is determined by bubble residence time and is shown in (3.61) [68].

$$S_{\varepsilon_l} = \frac{C_3 S_{k_l}}{t_b} \tag{3.60}$$

$$\omega_b = \frac{1}{t_b} = \left(\frac{2 C^{VM} D_d}{3 C_D \left| \boldsymbol{U}_r \right|} \right)^{-1} \tag{3.61}$$

The modeling of the lift force on particles perpendicular to the relative velocity is derived from Auton et al. [67] and shown in (3.62). C_L is the lift coefficient and it is most often set to a constant value of -0.025 when the bubbles are expected to depart from the surface into the bulk flow.

$$\boldsymbol{F}_L = C_L \alpha_v \rho_l \left[\boldsymbol{U}_r \times (\nabla \times \boldsymbol{U}_l) \right] \tag{3.62}$$

The basic wall dryout area fraction model is employed to specify the amount of heat flux applied at the wall is directed into the vapor convection instead of the forced convection, quenching, and evaporation of the liquid. This only occurs when the vapor volume fraction near the wall reaches a high value, known as the wall dryout breakpoint, so that it prevents liquid from coming into contact with the wall [55]. The wall dryout breakpoint (α_{dry}) is set at 0.9 so that once this vapor volume is reached, heat transfer to the vapor phase occurs and the wall boiling formulation is modified to include this effect and is shown in (3.63). This wall dryout breakpoint was selected based on experience gained from the PWR Subchannel and Bundle Test (PSBT) benchmark case [64], which is described in more detail in Section 3.4. The wall dryout breakpoint only affects the results of simulations when extremely high void fractions are reached in the cells near the wall because it is effectively modeling the heat transfer in dryout conditions.

$$q''_{tot} = (q''_{fc} + q''_q + q''_e)(1 - K_{dry}) + K_{dry} q''_{dry} \tag{3.63}$$

K_{dry}, as shown in (3.64), represents the effective wall contact area fraction of the vapor and is employed only above the dryout breakpoint. The term β is given in (3.65), where α_v is the void fraction of the vapor phase. The vapor convection is shown in (3.66) where h_{vc} is the vapor heat transfer coefficient. The CHF relaxation factor is set equal to 0.5 and is used to help with convergence if the dryout criterion is reached. It is specifically used for the update of the vapor volume fraction used in the dryout model.

$$K_{dry} = \beta^2 (3 - 2\beta) \tag{3.64}$$

$$\beta = \frac{\alpha_v - \alpha_{dry}}{1 - \alpha_{dry}} \tag{3.65}$$

$$q''_{dry} = h_{vc}(T_w - T_v) \tag{3.66}$$

While the current wall boiling 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 [48] which has been shown to occur in subcooled flow boiling through experiments [69, 70]. 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 [40, 48].

The standard wall boiling models are summarized in Table 3.5. 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 3.5. Standard Wall Boiling Models

Parameter	Model
Bubble Departure Frequency	Cole
Bubble Departure Diameter	Tolubinsky-Kostanchuk
Active Nucleation Site Density	Hibiki-Ishii
Bubble Influence Area	Kurul-Podowski (2.0)
Quenching Wait Coefficient	Del Valle Kenning (0.8)
Momentum closure	Symmetric Interaction Area Density
Drag Coefficient	Tomiyama
Lift Coefficient	-0.025
Turbulent Dispersion	Pr_τ = 1.0
Interaction Length Scale	Constant (1.0 mm)
CHF Relaxation	0.5
CHF Dryout Breakpoint	0.9
Condenation- Fluid	Chen-Mayinger
Condensation- Vapor	Constant Nu = 2.0
Interface Temperature	T_{sat}

3.4 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 Nuclear Power Engineering Cooperation (NUPEC) PBST International Benchmark exercise by Lo and Osman [64] 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+. The CAD models of the spacers were created using Autodesk Inventor and included the springs and dimples of the spacers. They were imported via parasolid files into STAR-CCM+ and rods were added to go through each spacer. The computational grid was created using unstructured polyhedral meshes in the rod-spacer sections and is shown in Figure 3.2. The contacts between the springs and dimples was modeled as shown in Figure 3.3.

The final rod bundle assembly, including 7 mixing vane spacers, 2 nonmixing vane spacers, and 8 simple spacers, connected the rod and spacer sections by extruding cells from the section ends. To decrease the total number of cells in the model, the extrusion expanded the cell height as the cells increased in distance from the section.

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^2-hr, the power 2990 kW (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. A depiction of the void fraction distribution downstream of a mixing vane spacer is shown in Figure 3.4.

The void fraction was averaged over the central 4 subchannels at a location of z = 3.177 m and the experimental value was 0.1791. The computed average over these 4 subchannels was 0.1576 [64]. This shows that current methods can provide reasonable void fraction results, but it is important to note that they have

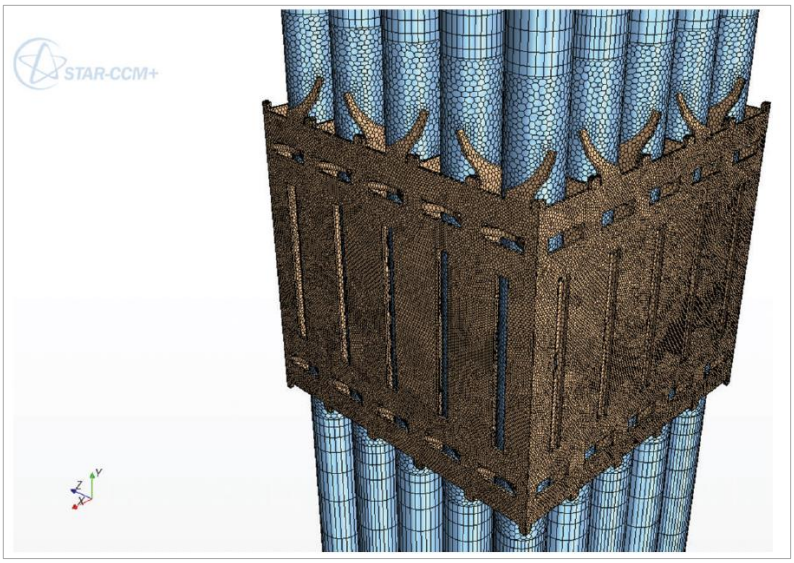


Figure 3.2. CFD grid of spacer with mixing vanes [64].

Figure 3.3. Rod and spacer contacts included in the model [64].

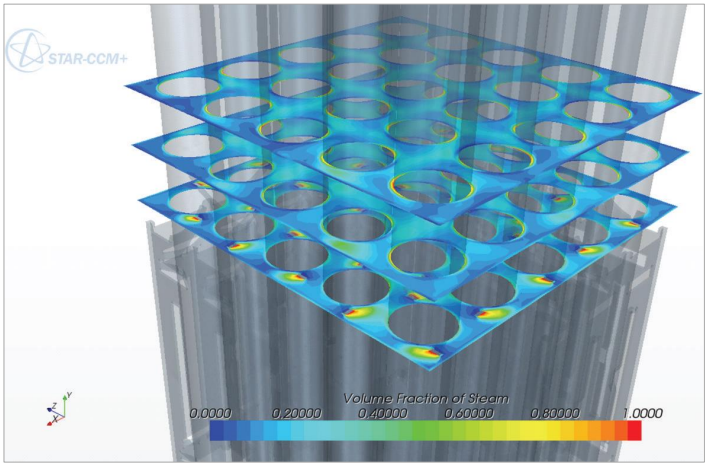


Figure 3.4. Void fraction downstream of a mixing vane spacer [64].

been empirically adjusted to do so. The goal of the current work is to provide a more general model that uses a mechanistic and physics-based approach and can also predict wall temperatures.

4 STAR-CCM+ CFD Software

4.1 STAR-CCM+ Capabilities

STAR-CCM+® is a commercial engineering software produced by CD-adapco™ that is used for solving fluid and solid flows, heat transfer, and stress problems and is the software used in the testing and validation of the new boiling model. The package contains a 3D-CAD modeler, automatic meshing, physics and turbulence modeling, and post-processing capabilities. It is based on programming technology that uses an object-oriented approach. The interface provides an object tree where the user defines the mesh characteristics and inputs the data and boundary conditions of the simulation [55].

Using the parametric solid 3D-modeler, geometries can be built from scratch. The 3D-CAD model can also be modified outside of the 3D-CAD which allows for modifications to be made quickly. An embedded CAD model can also be used if desired.

A surface wrapper can be employed that maps a high-quality triangulated mesh onto any surface. The user can enhance the the surface mesh by using surface-repair tools. Then the volume mesh is built with mostly hexahedral cells and can be created smoothly over multiple CAD parts. The mesh can also be extruded in any direction with control over the size, growth rate, and number of cells in the extrusion. If a turbulence model is used with a wall y^+ treatment, the prism layer mesh is manually controlled to produce the desired wall y^+ value [55].

The physics modeling capabilities of STAR-CCM+ include segregated, coupled, and finite volume solid stress solvers. The time portion of the simulation can be steady state, implicit and explicit unsteady, or harmonic balance. The turbulence modeling can be done using Reynolds-averaged Navier Stokes (RANS), large eddy simulation (LES), or laminar-turbulent transition. Fluids can be either incompressible or compressible and gases can be ideal or real. The heat transfer options include conjugate heat transfer, thermal surface to surface radiation, solar radiation, and discrete ordinates radiation. For simulating multi-phase, the user has Lagrangian particle tracking, Volume of Fluid (VOF), cavitation and boiling, Eulerian Multi-Phase (EMP), and melting and solidification options. Chemical reactions can be accounted for using reaction kinetics, eddy break-up (EBU), complex chemistry, and ignition options [55].

4.2 Solution Algorithms

A discretization method must be used to approximate the differential equations, using algebraic equations, that can be solved to obtain a numeric solution. To obtain a solution at a discrete location in space and time, the approximations are used in small space and time domains. Therefore, the solution is dependent on the quality of the discretization method [50].

Many discretization methods have been developed, but the most common are finite difference (FD), finite volume (FV), and finite element (FE). The STAR-CCM+ software employs the finite volume method [55]. In the FV method, the conservation equations are used in their integral form as a starting point. The domain being modeled is divided into a finite number of control volumes (CVs), which is where the conservation equations are applied. The variable values are calculated at the centroid (center) of each CV. The values calculated at the centroid are interpolated to the surfaces of the CV. A suitable quadrature formula is then used to approximate the surface and volume integrals [50].

The benefit of using a FV method is that it is suitable for complex geometries. The disadvantage of this method is that it is more difficult to develop methods of order higher than second in 3D as compared to the FD approach. STAR-CCM+ typically employs second-order accuracy in time and space [50, 55].

Multiple interpolation methods exist, and those available in STAR-CCM+ include first-order upwind, second-order upwind, central-differencing and bounded central-differencing [55]. The upwind differencing scheme (UDS) approximates the fluxes on the faces of the CVs by the value at the node upstream from the surface.

The flux ϕ_e is approximated as shown in (4.1), where $\boldsymbol{v}$ is the variable being calculated and $\boldsymbol{n}$ is the unit normal vector to face e. This is illustrated in Figure 4.1 where the face of interest (e) is highlighted in blue and centroids P and E are labeled.

$$\phi_e = \begin{cases} \phi_P, & \text{if } (\boldsymbol{v} \cdot \boldsymbol{n})_e > 0; \\ \phi_E, & \text{if } (\boldsymbol{v} \cdot \boldsymbol{n})_e < 0. \end{cases} \tag{4.1}$$

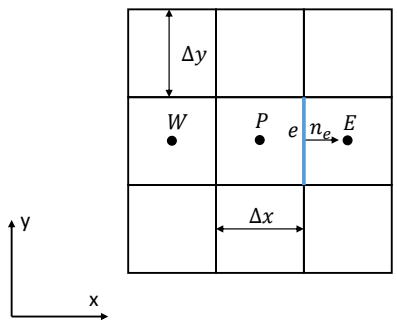


Figure 4.1. A typical CV layout with notation used for a 2-D Cartesian grid [50].

A series expansion is completed about centroid P when $(\boldsymbol{v} \cdot \boldsymbol{n})_e > 0$. An example of a Taylor series expansion for the Cartesian grid is shown in (4.2), where H represents the higher-order terms. A first-order upwind method retains only the first term on the right-hand side of the equation, so its leading truncation error represents a diffusive flux. The second-order upwind method uses the first two terms on the right-hand side of the equation by reconstructing the derivative of the flux using an upwind linear interpolation between the two upwind centroids.

$$\phi_e = \phi_P + (x_e - x_P)\left(\frac{\partial \phi}{\partial x}\right)_P + \frac{(x_e - x_P)^2}{2}\left(\frac{\partial^2 \phi}{\partial^2 x^2}\right)_P + H \tag{4.2}$$

The central-differencing scheme is also second-order and approximates the derivative using the two nearest nodes. The Taylor expansion for this method about the point x_P is shown in (4.3) where λ_e is defined in (4.4). This method uses the first two terms on the right-hand side of the series expansion shown in (4.3) [50].

$$\phi_e = \phi_E \lambda_e + \phi_P(1 - \lambda_e) - \frac{(x_e - x_P)(x_E - x_e)}{2}\left(\frac{\partial^2 \phi}{\partial^2 x^2}\right)_P + H \tag{4.3}$$

$$\lambda_e = \frac{x_e - x_P}{x_E - x_P} \tag{4.4}$$

Therefore, the approximation of the gradient between the two nodes P and E is shown in (4.5) for the assumption of a linear profile.

$$\left(\frac{\partial \phi}{\partial x}\right)_e = \frac{\phi_E - \phi_P}{x_E - x_P} \tag{4.5}$$

The bounded central-difference scheme is shown in (4.6) and is a function of the first-order upwind, the second-order upwind, and the central-differencing schemes represented by ϕ_{fou}, ϕ_{sou}, and ϕ_{cd} respectively.

ζ represents the Normalized-Variable Diagram (NVD) value that is based on local conditions. A smooth function of ζ is shown in (4.7) and must satisfy $\sigma(0) = 0$ and $\sigma(\zeta) = 1$ when $\zeta_{ubf} \leq \zeta$. The upwind blending factor is represented by ζ_{ubf} to provide balance between accuracy and robustness of the bounded central-differencing scheme. The larger values of ζ_{ubf} promote robustness while small values increase the accuracy [55].

$$\phi_e = \begin{cases} \phi_{fou}, & \text{for } \zeta < 0 \text{ or } 1 < \zeta; \\ \sigma\phi_{cd} + (1-\sigma)\phi_{sou}, & \text{for } 0 \leq \zeta \leq 1. \end{cases} \tag{4.6}$$

$$\sigma = \sigma(\zeta) \tag{4.7}$$

When unsteady problems are simulated, time must also be discretized and the solution advances in time step-by-step. STAR-CCM+ has the option of implicit or explicit for the psuedo-time integration. The explicit calculation is made assuming that all fluxes and sources are known at a time t_n. Then, at the next time level t_{n+1}, the value at a specific node is calculated using the solutions of the previous time level for all neighboring CVs. Therefore, each value is explicitly calculated for each node using the previous time step solution. The implicit method evaluates all the fluxes and source terms at the new time level in terms of the unknown variable values. This results in a system of algebraic equations that are similar to steady-state problems but have an additional contribution resulting from the unsteady term. The implicit method is useful when studying flows with slow transients [50, 55].

In STAR-CCM+, the semi-implicit method for pressure-linked equations (SIMPLE) algorithm is employed to reach the final single-phase solution. The steps this algorithm employs at each solution update until final solution convergence are [55]:

1. Set boundary conditions
2. Compute the reconstruction of velocity and pressure gradients
3. Calculate the velocity and pressure gradients
4. The discretized momentum equation is solved, creating an intermediate velocity field
5. The uncorrected mass fluxes at the faces are computed
6. The pressure correction equation is solved
7. The pressure field is updated using an under-relaxation factor for the pressure
8. The pressure boundary corrections are updated
9. The face mass fluxes are corrected
10. The cell velocities are corrected
11. The density is updated due to the pressure changes
12. Temporary storage is freed

A schematic of the flow solver SIMPLE algorithm is shown in Figure 4.2, where CGSTAB stands for the conjugate gradient squared stabilized algorithm developed by Van den Vorst and Sonneveld [71]. This is a more robust and stable method for solving systems of equations that are not necessarily symmetric. AMG refers to the algebraic multigrid method for solving the discrete linear system iteratively [55].

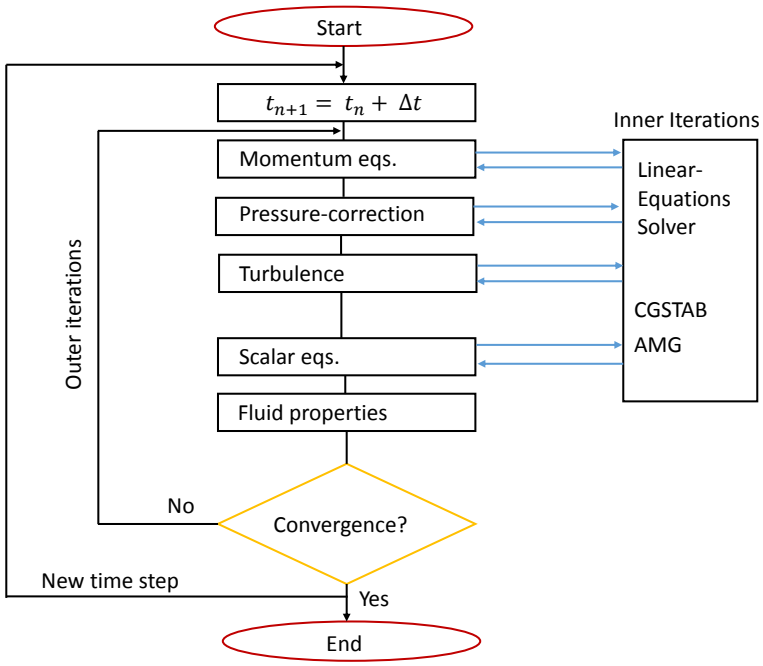


Figure 4.2. The SIMPLE scheme used in an iterative flow solver for single-phase [55].

CHAPTER TWO: NEW WALL BOILING MODEL DESCRIPTION

The new model is developed based on physical phenomena captured in subcooled flow boiling experiments; in particular, the phenomena that occur on the heated surface. This new model is a more general boiling closure model for subcooled flow boiling conditions that includes all physical boiling phenomena and necessary mechanisms to accurately predict the temperature and heat flux at the wall in CFD simulations. The model explicitly tracks the bubbles and dry area on the surface by assuming random distributions of the active nucleation sites. This information can be valuable in determining bubble interaction and total dry surface area on the surface.

5 Nucleation Site Density Modification

The new wall boiling model requires an accurate model for the prediction of the number of nucleation sites on the boiling surface, so a study was completed on available nucleation site density models to determine the model that most closely captures experimental trends. In particular, the Lemmert-Chawla [5], Hibiki-Ishii [42], and the Kocamustafaogullari-Ishii [38] models were investigated. The Lemmert-Chawla model is not dependent on pressure and is shown in (3.32) with m and p set as 185.0 sites/m^2 and 1.805 respectively. The Hibiki-Ishii model is dependent on the system pressure and the contact angle for the system and is shown in (3.34). It is applicable over a wide range of conditions, including mass velocities up to 886 kg/m^2-s, contact angles of 5-100°, and pressures up to 19.8 MPa.

The Kocamustafaogullari-Ishii model is dependent on the system pressure and also the contact angle through the use of the Fritz bubble departure diameter model [72, 73]. The nucleation site model is shown in (5.1) where N''^* is given in (5.2) with $f(\rho^*)$ and R_c^* shown in (5.3) and (5.4) respectively.

$$N'' = N''^*/D_d^2 \tag{5.1}$$

$$N''^* = f(\rho^*)R_c^{*-4.4} \tag{5.2}$$

$$f(\rho^*) = 2.157\text{x}10^{-7}\rho^{*-3.2}(1 + 0.0049\rho^*)^{4.13} \tag{5.3}$$

$$R_c^* = \frac{R_c}{D_d/2} \tag{5.4}$$

The bubble departure diameter D_d employed in this model is determined from the Fritz bubble departure correlation with a modification. The original Fritz correlation is shown in (5.5), where θ is in degrees, with the final calculation for D_d given by (5.6).

$$D_{dF} = 0.0148\theta\sqrt{\frac{2\sigma}{g\rho_l - \rho_v}} \tag{5.5}$$

$$D_d = 0.0012\rho^{*0.9}D_{dF} \tag{5.6}$$

The Kocamustafaogullari-Ishii model also includes the use of an effective wall superheat (ΔT_e) to account for the difference in the temperature gradient caused by the hydrodynamic flow field and is shown in (5.7). During forced convective nucleate boiling, the thermal boundary layer is much thinner, and thus the temperature gradient is much steeper than a pool boiling case with the same wall superheat. This causes the effective wall superheat to be lower than the true wall superheat. The suppression factor S is shown

in (5.8) and is calculated using the two-phase flow Reynolds number ($\mathrm{Re_{TP}}$), which is given in (5.9), where D_h is the hydraulic diameter, shown in (5.10) with A_f being the flow area and P_h the heated perimeter. The vapor quality is represented by x and the model is dependent on the speed of the flow through the inclusion of the mass flux (G) term.

$$\Delta T_e = S \Delta T_{sat} \tag{5.7}$$

$$S = \frac{1}{1 + 1.5\mathrm{x}10^{-5}\mathrm{Re_{TP}}} \tag{5.8}$$

$$\mathrm{Re_{TP}} = \frac{G(1-x)D_h}{\mu_l} \tag{5.9}$$

$$D_h = \frac{4A_f}{P_h} \tag{5.10}$$

The Lemmert-Chawla, Kocamustafaogullari-Ishii (at G = 500 kg/m^2-s), and Hibiki-Ishii models were compared at a pressure of 1.05 bar and assuming a contact angle of 100° and is shown in Figure 5.1. The contact angle was chosen assuming ITO is the boiling surface and water is the fluid. Experimental data [26, 41, 42, 74–76] has shown that the Hibiki-Ishii model most closely captures the sharp increase in nucleation site density at higher wall superheat. The influence of pressure was also investigated for the Hibiki-Ishii model and is shown in comparison to the Lemmert-Chawla model in Figure 5.2. This demonstrates that a nucleation site density model that is dependent on the system pressure is necessary since experiments have also illustrated the large effect of pressure on the nucleation site density.

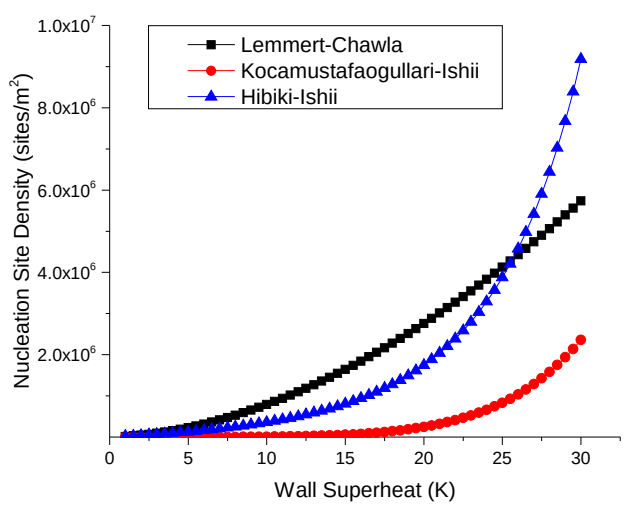


Figure 5.1. Comparison of the Lemmert-Chawla, Kocamustafaogullari-Ishii, and Hibiki-Ishii nucleation site density models at 1.05 bar.

A consequence of using a nucleation site density model that captures the sharp increase in the number of nucleation sites, such as the Hibiki-Ishii model, is that it can predict nucleation site densities of such high values that they are unphysical. To prevent the use of these unphysical values, a statistical bubble tracking method was developed and implemented by calculating the probability that a bubble forms underneath a bubble already present on the heated surface. Figure 5.3 illustrates the position of a new nucleation site

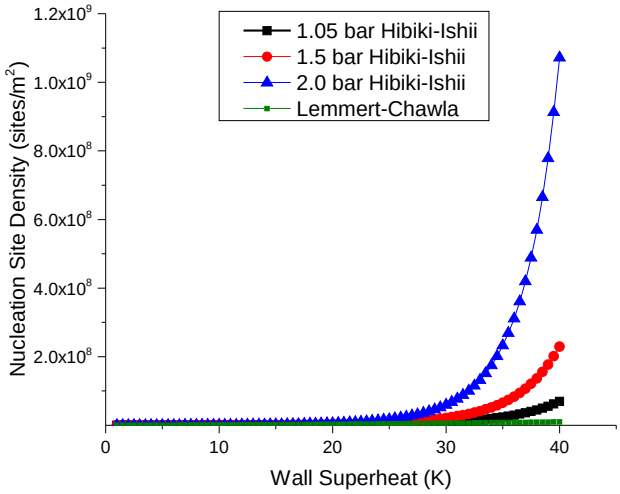


Figure 5.2. Comparison of the Hibiki-Ishii model at three different pressures to the Lemmert-Chawla model.

underneath a bubble already formed. Complete spatial randomness methods (CSR), which assumes that bubbles are randomly distributed on the heater surface, are employed to calculate the probability of this occuring. The nearest-neighbor method [77, 78] was used to determine the probability of one bubble center being within one bubble radius of another bubble center, and is shown in (5.11).

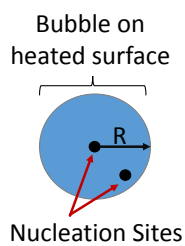


Figure 5.3. Illustration of a nucleation site that is underneath a bubble already present on the heated surface.

$$P = 1 - e^{-N_b'' \pi (D_d/2)^2} \tag{5.11}$$

The value represented by N_b'' must be used in place of the total nucleation site density to account for the fact that not all nucleation sites are active at the same point in time. To determine the number of nucleation sites active at a given time, the ratio of the growth time over the bubble period is used and the calculation is shown in (5.12), which assumes the bubbles nucleate with a temporally uniform distribution. The growth time (t_g) represents the amount of time the bubble is present on the surface and is calculated using the same bubble growth equation that is used in the force-balance bubble departure model. The calculation uses Zuber's growth model [79], which is the same model used by Yun [80] in his force balance model for bubble departure, but neglecting the component that accounts for subcooling. Therefore, since the bubble departure diameter is known through the force-balance prediction, the growth time is calculated as shown in (5.13) where b is a constant set equal to 1.56. The liquid thermal diffusivity is represented as η_l and is calculated as $k_l/(\rho_l c_{pl})$.

$$N_b'' = \left(\frac{1}{t_w + t_g}\right) t_g N'' = f t_g N'' \tag{5.12}$$

$$t_g = \frac{\pi (D_d/2)^2}{4b^2 \mathrm{Ja}^2 \eta_l} \tag{5.13}$$

The employment of this probability method enforces that the nucleation site density value used does not exceed the physical maximum number of bubbles that can fit on the heater surface. The modified nucleation site density is calculated as $(1 - P)N''$ and is employed as the nucleation site density value in the new partitioning model. An illustration of the effect of using this probability method is shown in Figure 5.4 at a pressure of 1.05 bar and a contact angle of 100° for both the Lemmert-Chawla and Hibiki-Ishii nucleation site density models. A bubble departure diameter of 1 mm was used in the calculation. Figure 5.4 shows how this probability method has the greatest effect on the nucleation site density models that have a sharp increase in the their predictions, such as the Hibiki-Ishii model. The Lemmert-Chawla model has a more gradual increase in the nucleation site density and therefore the physical maximum is not reached even at wall superheat of 40K.

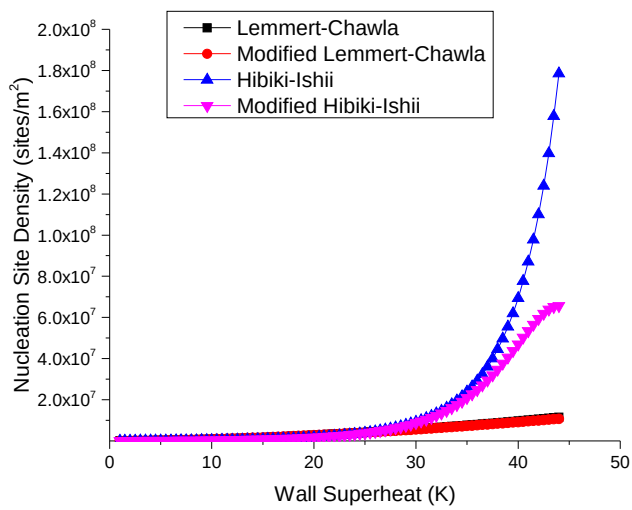


Figure 5.4. Illustration of the effect of the probability method on the nucleation site density.

6 Wall Boiling Model Formulation

The proposed model aims to capture all of the boiling phenomena that occur during subcooled flow boiling to accurately predict the temperature and heat flux for subcooled flow boiling at the wall in CFD simulations. As a result, the wall heat flux (q''_{tot}) is partitioned into four components that capture the physical phenomena evident from experimental data: forced convection (q''_{fc}), quenching (q''_q), evaporation (q''_e), and sliding conduction (q''_{sc}) and is shown in (6.1). The phenomena the components capture are depicted in Figure 6.1. 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. This sliding conduction term captures all the heat transfer that is associated with disruption and reformation of the thermal boundary layer that is caused by the growth and movement of bubbles. These four modes of heat transfer are discussed in detail in the following sections.

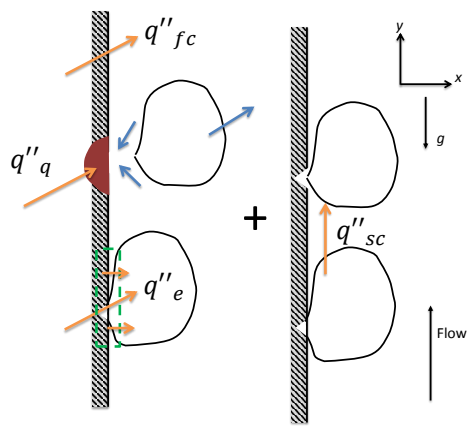


Figure 6.1. Depiction of the heat flux partitioning for subcooled flow boiling.

$$q''_{tot} = q''_{fc} + q''_q + q''_e + q''_{sc} \tag{6.1}$$

6.1 Forced Convection Component

The forced convection heat flux is calculated using the traditional CFD method of single-phase forced convection on a heated surface in turbulent flow and the local calculation is shown in (6.2). The non-dimensional temperature term for the liquid (t^+) is defined by the thermal wall law for temperature, a mathematical description of the temperature profile in the turbulent boundary layers. The wall law is chosen automatically by the STAR-CCM+ software because it is based on the turbulence mode behavoir and it is calculated using the velocity profile near the wall [55]. The friction velocity (u_τ) is given in Table 6.1.

$$q''_{fc} = \frac{\rho_l c_{pl} u_\tau}{t^+}(\Delta T_{sup} + \Delta T_{sub}) \tag{6.2}$$

The new boiling model's forced convection component differs from the baseline 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 completed 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.

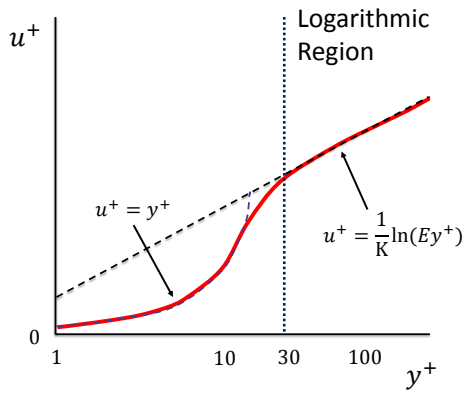


Figure 6.2. Velocity profile near the wall.

For a hydraulically smooth wall, the velocity profile near the wall is given by (6.3) for high-Reynolds number flows inside the viscous-dominated region of the boundary layer that is not resolved. An illustration of the velocity profile near the wall is shown in Figure 6.2. The non-dimensional velocity (u^+) is a function of the non-dimensional distance from the wall (y^+). The empirical coefficient that is the log-law offset value (E) is usually set equal to 9.0 and is a constant from the rearrangement of the classic equation (6.4) to give $E = e^{\kappa B}$, where κ = 0.42 is the Von Karman constant. The definitions of the remaining variables are listed in Table 6.1, where τ_w is the wall shear stress. The viscous sublayer outside of the logarithmic region occurs for a y^+ from 0 to y_m^+ [55, 81].

$$u^+ = y^+ \text{ when } y^+ \leq y_m^+ \qquad (6.3)$$
$$u^+ = \frac{1}{\kappa} ln(Ey^+) \text{ when } y^+ > y_m^+$$

$$u^+ = \frac{1}{\kappa} ln(y^+) + B \qquad (6.4)$$

Table 6.1. Definition of variables used in the smooth wall functions.

Variable	Definition
u^+	$\frac{u}{u_\tau}$
u_τ	$\sqrt{\frac{\tau_w}{\rho_l}}$
y^+	$\frac{\rho_l u_\tau y}{\mu_l}$

To account for the added roughness caused by the bubbles on the heater wall, the rough wall model is employed [55, 82]. This model modifies the log-law coefficient E to make it a function of a roughness parameter given by R^+. This is shown in (6.5), 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 (6.4). The roughness function is dependent on the value of the roughness parameter and is shown in (6.6).

$$R^+ = \frac{r u_\tau \rho_l}{\mu_l} \tag{6.5}$$

$$f = \begin{cases} 1 & \text{when } R^+ \leq R^+_{smooth} \\ \left[B \left(\frac{R^+ - R^+_{smooth}}{R^+_{rough} - R^+_{smooth}} \right) + C R^+ \right]^a & \text{when } R^+_{smooth} < R^+ < R^+_{rough} \\ B + C R^+ & \text{when } R^+ > R^+_{rough} \end{cases} \tag{6.6}$$

In the intermediate range of wall roughness, where $R^+_{smooth} < R^+ < R^+_{rough}$, a is given by (6.7). Typically, the values for R^+_{smooth}, R^+_{rough}, B, and C are 2.25, 90, 0, and 0.253 respectively [82].

$$a = \sin \left[\frac{\pi}{2} \frac{\log \left(R^+ / R^+_{smooth} \right)}{\log \left(R^+_{rough} / R^+_{smooth} \right)} \right] \tag{6.7}$$

To predict the added roughness due to the bubbles, the equivalent sand-grain 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 [83], the value is estimated by calculating an average surface roughness (R_a) due to the bubbles. This is shown in (6.8) where n is the number of measurements and y_i is the height for each measurement.

$$R_a = \frac{1}{n} \sum_{i=i}^{n} |y_i| \tag{6.8}$$

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 and is depicted in Figure 6.3 on a heated surface with hemispherical bubbles. The bubbles are assumed to be evenly distributed in a square lattice so that the bubble spacing (s) is given by $s = 1/\sqrt{N''}$. The average surface roughness is then calculated using an integral form of (6.8) and is shown in (6.9) 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 (6.10), where R_d is the bubble departure radius.

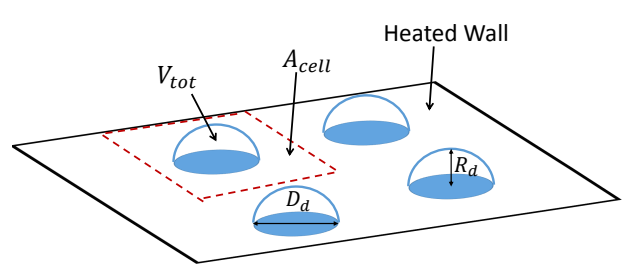


Figure 6.3. Depiction of the unit cell and volume taken up by a hemispherical bubble.

$$R_a = \frac{V_{tot}}{A_{cell}} \tag{6.9}$$

$$V_{tot} = \frac{4}{3}\frac{1}{2}\pi R_d^3 + \pi R_d^3 = \frac{5}{3}\pi R_d^3 \tag{6.10}$$

An example to illustrate the effectiveness of this method was completed by implementing the results from the ITM-1 study [84]. 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 CFD code TransAT [85]. 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 6.4, 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, which was calculated as shown in (6.11), a fixed density (ρ) of 1kg/m^3, and viscosity (μ) of Re_τ^{-1}.

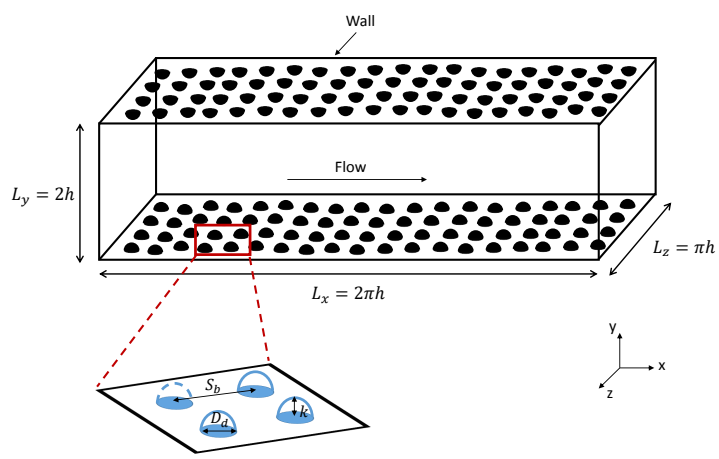


Figure 6.4. ITM Test Case 1 computational domain and illustration of the hemispherical obstacles [84].

$$\mathrm{Re}_\tau = \frac{h\rho u_\tau}{\mu} \tag{6.11}$$

The average surface roughness was calculated using the integral from described previously. In this test case, the bubbles were modeled as hemispheres, so $V_{tot} = 2/3\pi R_d^3$. This provided $R_a = 3.27$ mm. 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 from Kasagi [86] 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 6.5. This is important for forced convective heat transfer because the temperature profile is then obtained from this velocity profile.

The single-phase forced convection component is employed on all areas of the heater that are not covered by a bubble. The mathematical implementation of the forced convection component on only the areas unaffected by a bubble is detailed in Section 8. The calculation for the forced convection component is given by (6.12), where h_{fc} is the heat transfer coefficient that incorporates the increased heat transfer due to the presence of bubbles when wall boiling occurs. This is completed by relating the flow parameters to the thermal parameters, which is used to infer the heat transfer given that the momentum transfer is known.

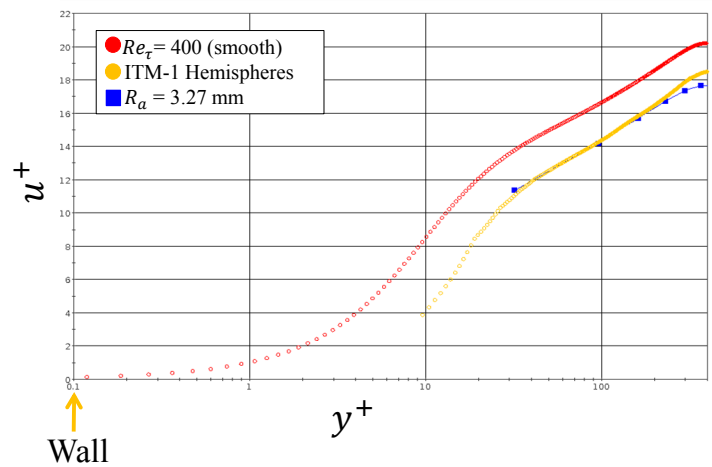


Figure 6.5. Velocity profile near the wall of the rough wall function (blue) compared to the DNS data of the smooth (red) and hemispherical obstacle (yellow) surfaces.

For this reason, the affected velocity profile from the increased surface roughness causes an increase in the forced convection heat transfer coefficient.

$$q''_{fc} = h_{fc}(\Delta T_{sup} + \Delta T_{sub}) \tag{6.12}$$

6.2 Sliding Conduction Component

As experiments have illustrated [28], the effect of sliding bubbles on the heat transfer coefficient for sub-cooled 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. [87] and Zeng et al. [20, 88] and it was recently adapted and modified by Yun et al. [80] for CFD application and further improved by Sugrue et al. [89]. The mechanistic bubble departure model implemented in this work is based on the model by Sugrue et al.

The 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 based on the work by Sugrue et al. [89], which is a modified version of that developed by Yun. This model based on the work by Sugrue et al. is used to predict the bubble departure diameter and if the bubble immediately lifts-off the surface or slides prior to lift-off. If the prediction of D_l is smaller than D_m, then the bubble lifts-off into the fluid flow without sliding. If the bubble slides along the heater surface before lift-off, the lift-off diameter (D_l) is predicted by a force-balance model based on that developed by Situ et al. [15]. The movement of a bubble from its inception point that then slides and lifts-off is illustrated in Figure 6.6.

Bubble growth and sliding increase heat transfer because of the disruption of the thermal boundary layer inducing transient conduction as the cold liquid comes in contact with the wall. The additional evaporation that occurs while the bubble slides is captured by the microlayer evaporation component described in Section 6.4. If the bubble lifts-off the surface without sliding, it also disturbs the thermal boundary layer and is therefore is captured in this term. The transient conduction is calculated using the error function solution to the transient temperature profile for 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 [40, 90]. The time-dependent wall heat flux for this solution is shown in (6.13). When the boundary layer is disrupted by a sliding bubble,

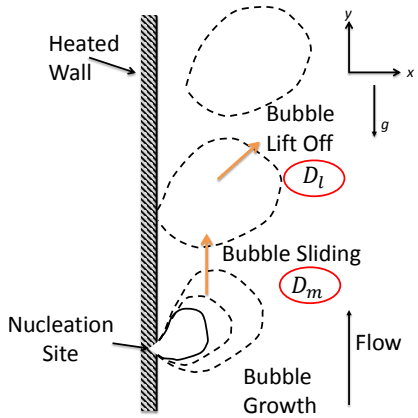


Figure 6.6. Illustration of bubble growth at a nucleation site until it departs from the site, slides, and lifts-off the surface.

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 heat transfer coefficient and is calculated as shown in (6.14). By integrating the error function solution over this time, the sliding conduction term is given by (6.15).

$$q''_w = \frac{k_l(T_w - T_l)}{\sqrt{\pi \eta_l t}} \tag{6.13}$$

$$t^* = \left(\frac{k_l}{h_{fc}}\right)^2 \frac{1}{\pi \eta_l} \tag{6.14}$$

$$q''_{sc} = \frac{2k_l(T_w - T_l)}{\sqrt{\pi \eta_l t^*}} a_{sl} t^* f N''^* \tag{6.15}$$

The sliding length (l) and area influenced by the sliding bubbles (a_{sl}) is determined by the rate of 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 6.7, 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.

The growth of the bubble while sliding is determined from a correlation against Maity's data [91] of a single bubble sliding in subcooled flow boiling and is shown in (6.16) 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. The $\mathrm{Re_b}$ is calculated as shown in (6.17).

$$\frac{(D_{sl}^2 - D_{in}^2)}{t_{sl}\eta_l \mathrm{Ja}} = \frac{1}{15(0.015 + 0.0023\mathrm{Re_b}^{0.5})(0.04 + 0.023\mathrm{Ja}^{0.5})} \tag{6.16}$$

$$\mathrm{Re_b} = \frac{D_d U_l \rho_l}{\mu_l} \tag{6.17}$$

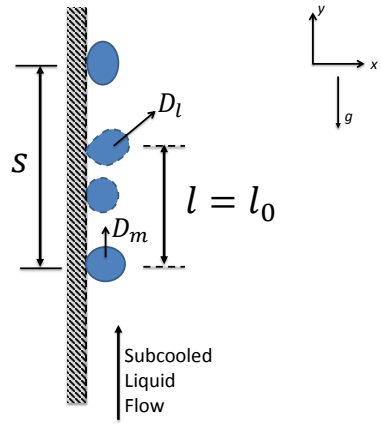


Figure 6.7. Illustration of a bubble sliding from its inception point and then departing from the heater after sliding a distance given by l.

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 (6.18). υ_b is the sliding velocity of the bubble, which is assumed to be equal to the liquid velocity surrounding the bubble, 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 [92, 93]. The absorption of a bubble at a second nucleation site is illustrated in Figure 6.8. The addition of the volume of the bubble swept by the original sliding bubble is assumed to be a bubble with diameter D_m. The diameter of the bubble after absorbing another bubble is therefore given by (6.19).

$$t_{sl} = \frac{s}{\upsilon_b} \tag{6.18}$$

$$D^* = \left(D_{sl}^3 + D_m^3\right)^{1/3} \tag{6.19}$$

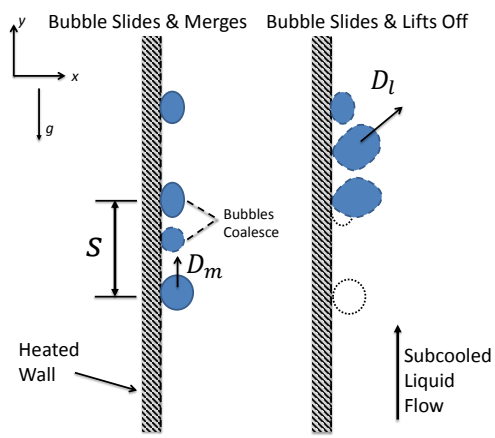


Figure 6.8. Illustration of a bubble sliding from its inception point and merging with another bubble after sliding a distance s.

The total distance a bubble slides can then be calculated as shown in (6.20) 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 traveled from the last bubble merger to lift-off. Using the average bubble size during sliding (D_{avg}) as shown in (6.21), the area affected by a sliding bubble is then $a_{sl} = D_{avg} l$.

$$l = N_{merged} s + l_{D_n - D_l} \tag{6.20}$$

$$D_{avg} = D_m + 0.5(D_l - D_m) \tag{6.21}$$

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 for the calculation of the sliding conduction component. This is completed by employing a reduction factor (R_f), shown in (6.22), when the sliding distance is greater than the spacing between bubbles. This provides a new active nucleation site density value (N''^*) that is shown in (6.23).

$$R_f = \frac{1}{l/s + 1} \tag{6.22}$$

$$N''^* = R_f N'' \tag{6.23}$$

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. During vertical up-flow, the imposed pressure gradient drives spherical bubbles to remain on the walls and out of the center of the channel. When the bubbles are no longer spherical in shape, the shear lift force causes bubbles to move away from the walls and into the bulk flow [94, 95].

To account for the effect of bubble deformability in the prediction of the bubble lift-off diameter, the Eo number is used. A single bubble of light fluid rising in an unbounded flow is usually described by the Eo number and the Morton (Mo) number, which is shown in (6.24). The Mo number compares the viscous to the capillary forces. For a given fluid and pressure, the Eo number is a characteristic of the bubble size and the Mo number is a constant. At low Eo number, a bubble is spherical. At a higher Eo, it becomes ellipsoidal and possibly wobbly if the Mo number is low (which is usually the case in low viscosity liquids like water). At higher Eo numbers, the bubble adopts a spherical-cap shape with trailing skirts if the Mo number is high [96, 97].

$$\mathrm{Mo} = \frac{(\rho_l - \rho_v) g \mu_l^4}{\rho_l^2 \sigma^3} \tag{6.24}$$

A DNS study still in progress by Dabiri et al. investigated the regime transitions in vertical channel up-flow due to bubble deformability. The simulation was completed assuming the vapor density is one-tenth of the liquid density to reduce computational cost, and the fluid had a high Mo number. At low Eo, the bubbles were spherical and remained near the walls causing a lower flow rate than the single-phase solution. As the Eo number rose, the bubbles moved to a more uniform distribution and the flow rate increased to the single-phase solution. This transition occurred rather abruptly at an Eo number of 2.5 and when the Eo number was greater than 3.5, there were no bubbles sliding along the walls. For high Reynolds flows and an Eo number less 0.1, bubbles typically remain spherical. Usually, above an Eo of 0.1, the bubbles begin to deviate from the spherical shape [97]. In subcooled flow boiling, the bubbles can also lift-off the heated surface but still remain near the wall and not enter the bulk flow.

For very low viscosity liquids such as water, the Mo varies from 1.8x10^{-13}, at a pressure of 2 bar, to 2.3x10^{-12} at a pressure of 150 bar. Charts such as the Re number (shown in (6.25)) versus Eo number for different

Mo numbers [98, 99] can be used to describe the shape of the bubble. At these very low Mo numbers characteristic of water, the bubble deviates from spherical shape at very low Eo numbers. Clift et al. [98] also completed a study comparing the aspect ratio of the horizontal and vertical diameters of a bubble to the Eo number for different Mo numbers. The aspect ratio is defined as the maximum vertical dimension of the bubble over the maximum horizontal dimension of the bubble. A correlation was developed to relate the aspect ratio (E) to the Eo number of the bubble for $\mathrm{Mo} \leq 10^{-6}$, and is shown in (6.26).

$$\mathrm{Re} = \frac{\rho_l U_l D_h}{\mu_l} \tag{6.25}$$

$$\mathrm{E} = \frac{1}{1 + 0.163\mathrm{Eo}^{0.757}} \tag{6.26}$$

In the experimental data of Prodanovic et al. [28], the bubble lift-off diameters were recorded with the maximum vertical and maximum horizontal diameter of the bubble at the moment of lift-off at a pressure of 2 bar. The typical aspect ratio of the bubbles at the moment of lift-off was about 0.96. This provides an Eo value of approximately 0.1, which describes the maximum Eo number a bubble reaches before lifting-off the heated surface. Therefore, a maximum Eo number of 0.1 was implemented for up-flow simulations to capture when the bubbles begin to lift-off the surface.

6.3 Quenching Component

The quenching component in this new model captures the heat transfer associated with bringing the dry area of the bubble on the heater back to the wall superheat and temperature distribution prior bubble inception. Therefore, the quenching term becomes a function of the material properties of the heated surface. This is appropriate because the quenching component can capture the influence of the type of heated material on the heat transfer coefficient. This has been observed experimentally when the wall thermal capacity was seen to be an important correlating parameter. For example, the Klimenko correlation was developed to incorporate the wall thermal conductivity for prediction of the heat transfer coefficient [100].

The quenching component is employed when a bubble departs from its nucleation site and is depicted in Figure 6.9. The quenching heat flux is shown in (6.27) where ρ_h is the density of the heater material, c_{ph} is the specific heat capacity of the heater, ΔT_h is the average temperature difference between the hot spot on the heater and the surrounding wall temperature in the bubble location, and V_q is the volume of heater material contained in the hot spot and is given in (6.28). The volume of the hot spot is assumed to be of hemispherical shape that has the same diameter as the dry area. The dry area is assumed to have half the radius of the bubble because the microlayer evaporation is assumed to occur out to half the radius of the bubble as was shown through experimental work [11, 16, 101]. Experimental data by Gerardi et al. [16] provided evidence to approximate ΔT_h as 2 K, and a sensitivity study was completed on this value and detailed in Section 11.

$$q_q'' = \rho_h c_{ph} \Delta T_h V_q f N'' \tag{6.27}$$

$$V_q = \frac{2}{3}\pi \left(\frac{D_{dry}}{2}\right)^3 \tag{6.28}$$

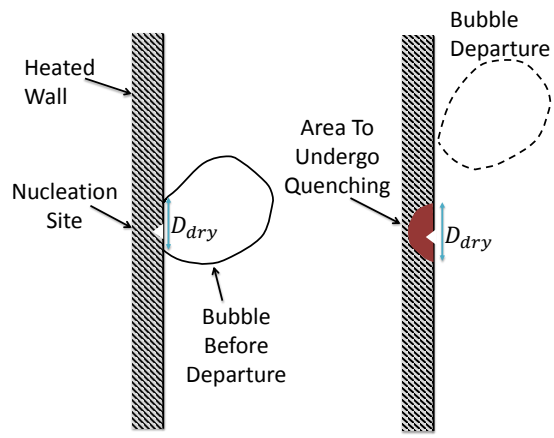


Figure 6.9. Illustration of bubble growth on a heater and the area of the bubble that is involved in the quenching heat flux component.

6.4 Evaporation Component

The evaporation term is calculated by capturing the physical phenomena of the rapid initial bubble growth and the microlayer evaporation and is shown in (6.29). The initial bubble growth ($q''_{e,init}$) is calculated as shown in (6.30). The microlayer evaporation location is illustrated in Figure 6.10.

$$q''_e = q''_{e,init} + q''_{e,mL} \tag{6.29}$$

$$q''_{e,init} = \frac{4}{3}\pi \left(\frac{D_d}{2}\right)^3 \rho_v h_{fg} f N'' \tag{6.30}$$

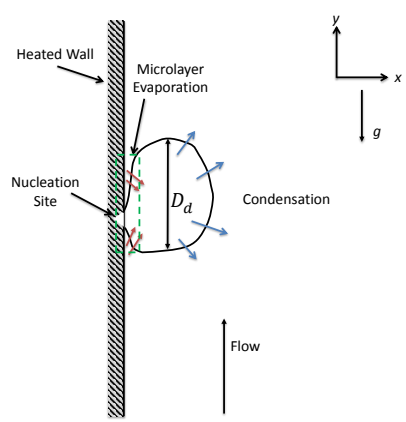


Figure 6.10. Illustration of microlayer evaporation and condensation on a bubble attached to a heated wall.

A closer look at the bubble microlayer is shown in Figure 6.11. The volume of the microlayer (V_{mL}) is calculated by assuming the microlayer has a maximum thickness (δ_{max}) that decreases linearly to zero at the center of the bubble and is shown in (6.31). It is assumed that the entire microlayer evaporates for each bubble. The calculation for the microlayer ($q''_{e,mL}$) component is given in (6.32). The value used for the

radius of the microlayer is half the radius of the bubble, as was shown by experiments by Gerardi [14]. The maximum microlayer thickness is $\delta_{max} = 2\,\mu\text{m}$, which is a typical value for bubble microlayer thickness approximation [102]. A discussion on the sensitivity of this maximum microlayer thickness is included in Section 11.

$$V_{mL} = \frac{2}{3}\pi \left(\frac{D_d}{4}\right)^2 \delta_{max} \tag{6.31}$$

$$q''_{e,mL} = V_{mL}\rho_f h_{fg} f N'' \tag{6.32}$$

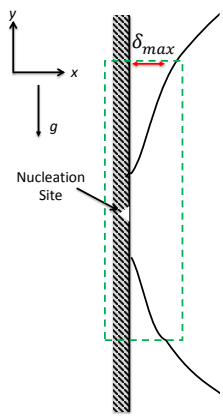


Figure 6.11. Depiction of the microlayer showing the maximum thickness.

7 Bubble Tracking, Merging, and Dry Surface Area

The number of bubbles on the heated surface at a given point in time is tracked to determine the amount of dry surface area on the heater. This may be a key factor in future testing for prediction of the critical heat flux since CHF is recognized by large patches of dry areas on the heated surface [11].

Experiments have shown that the solid-liquid-vapor triple contact line is highly irregular and dynamic during nucleate boiling [11]. The liquid menisci advance into and retreat from the dry patches as a function of time, due to the effects of bubble nucleation, liquid inertia (sloshing), capillary forces (surface tension) and recoil forces (evaporation). The wetted area fraction, $(1 - S_{dry})$, can have large fluctuations and this is sufficient to sustain a very high heat flux without experiencing CHF because no point on the surface is permanently dry due to liquid sloshing.

Bubbles can merge on the heater prior to departure from the surface. A statistical bubble merging tracking method was developed and implemented by calculating the probability that bubbles merge while on the heater surface. It employs the same CSR methods as the nucleation site density limiting approach, which assumes that bubbles are randomly distributed on the heater surface. The nearest-neighbor method was used to determine the probability of one bubble center being within one bubble diameter of another bubble center (see Figure 7.1).

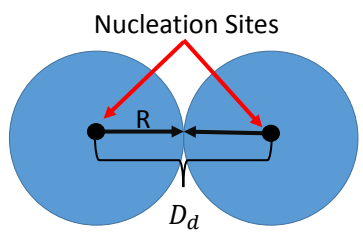


Figure 7.1. Illustration of the maximum distance between two bubble centers that allows for bubble merging.

Given the bubble departure size and the active nucleation site density at a given point in time ($N_b'' = ft_g N''$), the probability that a bubble will merge with its nearest neighbor is given by (7.1). This assumes that to merge, the bubbles must physically touch. An example of bubbles that would merge on the heated surface is depicted in Figure 7.2. The number of bubbles that merge indicates how the dry surface patches are distributed and their size. To calculate the amount of dry surface area, it is assumed that the dry areas of the merged bubbles also merge. Therefore, the fraction of dry surface area of the heater (S_{dry}) is calculated as shown in (7.2) where A_{dry} and A_h are the total dry and heater areas respectively.

$$P = 1 - e^{-N_b'' \pi (D_d)^2} \tag{7.1}$$

$$S_{dry} = \frac{A_{dry}}{A_h} = N_b'' \pi \left(\frac{D_d}{4} \right)^2 \tag{7.2}$$

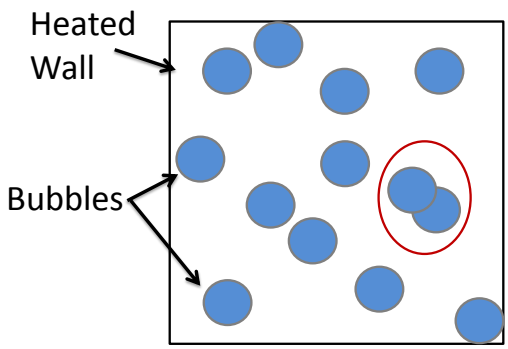


Figure 7.2. Illustration of two bubbles circled in red that would merge on the heated surface.

8 Model Implementation in STAR-CCM+ Software

8.1 User Code External Routine

The new boiling model is coded in C programming language on a Linux OS machine in files outside of the STAR-CCM+ software. Each subroutine is linked to STAR-CCM+ by user libraries that are compiled from the subroutines. Four files are used to build each library: uclib.h, uclib.c, ucmodels.h, and ucmodels.c, which are included in Appendix A. The file uclib.h defines the registration of the variables that will be used. The file uclib.c registers the application-specific variables for the usercode that will be shared with the remaining two files. The ucmodels.h file calls the variables specified from the simulation so that they may be implemented. The ucmodels.c file contains the coded model that uses the variables from the simulation, and returns the final value that can be used in the simulation. The force-balance bubble departure diameter model is also implemented through a user code.

8.2 Linkage to Software

The standard EMP baseline models are activated for the boiling simulations. The new boiling model is implemented through the heated region boundary. Under the Liquid and Phase Interaction phase conditions, the user wall heat flux coefficient specification is set to "specified." This allows the heat flux relationship at the wall to be controlled. Under the physics values, the boiling model is implemented through the A, B, and C coefficients. These coefficients are used in the linearized wall heat flux equation shown in (8.1), where q''_w is the heat flux applied to the wall. The equation given by (8.1) is employed for both (1) the heat transfer to the liquid phase that does not result in a phase change and will be referred to as the liquid coefficients, and (2) the heat transfer that is associated with the phase change from liquid to vapor that is referred to as the phase interaction coefficients.

$$q''_w = A + BT_l + CT_w \tag{8.1}$$

The model is coded such that all components multiplied by the liquid temperature are included in the coefficient B and all components that are multiplied by the wall temperature are included in coefficient C. All remaining heat transfer components to the liquid that are not explicitly dependent on the liquid or wall temperature are included in coefficient A.

The quenching and microlayer evaporation components are included in the liquid coefficient A and are set equal to zero if the wall temperature is not greater than the saturation temperature. Since coefficient A is not multiplied by a temperature, the units are simply that of the heat flux [W/m^2]. Therefore, the calculation for liquid coefficient A is shown in (8.2) where q''_q and q''_{mL} are calculated as shown (6.27) and (6.32).

$$A = -(q''_q + q''_{mL}) \tag{8.2}$$

The forced convection and sliding conduction terms are included in the liquid coefficients B and C. The units of both coefficients B and C are [W/m^2-K] and they can be described as heat transfer coefficients. Therefore, coefficient B is calculated as is shown in (8.3) where h_{sc} is the heat transfer coefficient associated with the sliding conduction term and is shown in (8.4). The liquid coefficient C is shown in (8.5).

$$B = h_{sc} + h_{fc} \tag{8.3}$$

$$h_{sc} = \frac{2k_l}{\sqrt{\pi \eta_l t^*}} a_{sl} t^* f N''^* \tag{8.4}$$

$$C = -(h_{sc} + h_{fc}) \tag{8.5}$$

The sliding conduction term only acts on the surface area under the influence of the bubbles, while the forced convection term acts on the remaining surface area. h_{fc} is calculated by using the internal B liquid coefficient from the simulation, represented by h_b, and is shown in (8.6). The forced convection over the heater area unaffected by bubbles is captured in h_{fc1}, which is shown in (8.7), and h_{fc2} represents the forced convection term that occurs over the sliding conduction area for the time in which the forced convection heat transfer coefficient is greater than the sliding conduction heat transfer coefficient and is shown in (8.8).

$$h_{fc} = h_{fc1} + h_{fc2} \tag{8.6}$$

$$h_{fc1} = h_b(1.0 - K_{quench})(1.0 - a_{sl}N''^*) \tag{8.7}$$

$$h_{fc2} = h_b(1.0 - K_{quench})a_{sl}N''^*(1.0 - ft^*) \tag{8.8}$$

The initial bubble inception evaporation term is included through the phase interaction coefficient A, which also has the units of [W/m^2] and is shown in (8.9). There are no components associated with the phase change that include an explicit dependence on the wall or liquid temperatures, so both the phase interaction coefficients B and C are equal to zero.

$$A = -\left[\frac{4}{3}\pi\left(\frac{D_d}{2}\right)^3 \rho_v h_{fg} f N''\right] \tag{8.9}$$

8.3 Field Functions

The STAR-CCM+ software has another method to implement initial conditions or boundary values through the use of field functions. In this work, user defined field functions were used to implement the nucleation site density (N''), the equivalent sand-grain roughness height to be used in the rough wall surface specification (r), and to calculate the fraction of dry surface area (S_{dry}).

The active nucleation site density was implemented through a field function to increase the stability of the simulation. The Hibiki-Ishii nucleation site density model was implemented as only a function of the wall superheat by creating an exponential curve that was specific for the pressure and the contact angle associated with each test case. The equivalent sand-grain roughness height was calculated as shown in (8.10). This field function was then selected as the roughness height for the rough wall surface specification. The fraction of dry surface area was calculated as shown in (7.2).

$$r = \frac{5}{3}\pi\left(\frac{D_d}{2}\right)^3 N''_b \tag{8.10}$$

The angle of the test section is a required input for the bubble departure and lift-off diameter predictions described in Section 9. To calculate the angle, a field function was created based on the direction of the gravity force in the simulation. First, a field function was created that defined the gravity vector in the x, y, and z-directions. An example of the gravity vector for a vertical upflow test section is shown in (8.11). The angle of the test section was then calculated using the direction of the wall normal and the direction of gravity. The code for these field functions is provided in Appendix B.

$$\boldsymbol{g} = [0, 0, -9.81] \tag{8.11}$$

9 Mechanistic Force-Balance Models

Force-balance models are used to predict both the departure diameter from the nucleation site and also the lift-off diameter from the surface. Many of the calculations for the heat flux partitioning are dependent on the bubble departure and lift-off diameters so these values are of high importance for the model. As a result, a mechanistic force-balance model based on the work by Sugrue et al. [23] is implemented in the code that modifies the original force-balance by Klausner et al. [87] and Zeng et al. [20] and the CFD implemented version by Yun et al. [80, 103]. This revised model by Sugrue was built by systematically investigating experimental data of bubble departure diameters over a range of mass fluxes, test section angle orientations, pressures, and subcoolings. Another force-balance model proposed by Situ et al. [15] and tested also by Khanlou et al. [27] incorporates the specific physical phenomena of a sliding bubble to better predict the lift-off diameter.

9.1 Bubble Departure Diameter

The mechanistic bubble departure model based on the work by Klausner et al. [87] and recently improved by Sugrue et al. [23] for flow boiling employs a summation of all the forces acting both parallel and perpendicular to a bubble on a heated surface. Once the summation of one 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 9.1 illustrates the forces calculated. This model is employed to predict the bubble diameter when the bubble leaves its nucleation site. If the sum of the forces in the y-direction are greater than zero (implemented in the code as $\Sigma F_y > 10^{-15}$ to account for computational error) before the x-direction, then no sliding occurs. If the sum of the forces in the x-direction are greater than zero ($\Sigma F_x > 10^{-15}$) before the y-direction, then the bubble slides along the heater surface until it reaches the lift-off diameter predicted by the model by Situ et al.

The summation of the forces acting on the bubble are calculated with a dependence on the orientation of the heater, and the equations are shown in (9.1) and (9.2) for the x- and y-directions respectively. The orientation angle of the heater surface is represented by ϑ, 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). The summation in the x-direction includes the surface tension force (F_{sx}), the quasi-steady drag force (F_{qs}), the buoyancy force (F_b), and the unsteady drag force (F_{du}). The summation in the y-direction includes the surface tension force (F_{sy}), the shear lift force (F_{sL}), the buoyancy force, the hydrodynamic pressure force (F_h), the contact pressure force (F_{cp}), and the unsteady drag force.

$$\Sigma F_x = F_{sx} + F_{qs} - F_b \sin\vartheta + F_{du} \sin\phi \tag{9.1}$$

$$\Sigma F_y = F_{sy} + F_{sL} - F_b \cos\vartheta - F_h + F_{cp} + F_{du} \cos\phi \tag{9.2}$$

The surface tension forces are dependent on the advancing (α) and receding (β) contact angles of the system (in radians) and are shown in (9.3) and (9.4) for the x- and y-directions respectively. The bubble foot diameter captures the surface contact diameter of the bubble and is represented by d_w and is calculated as shown in (9.5).

$$F_{sx} = -1.25 d_w \sigma \frac{\pi(\alpha - \beta)}{\pi^2 - (\alpha - \beta)^2} \tag{9.3}$$

$$F_{sy} = d_w \sigma \frac{\pi}{\alpha - \beta} [\cos\beta - \cos\alpha] \tag{9.4}$$

$$d_w = 0.025 D_d \tag{9.5}$$

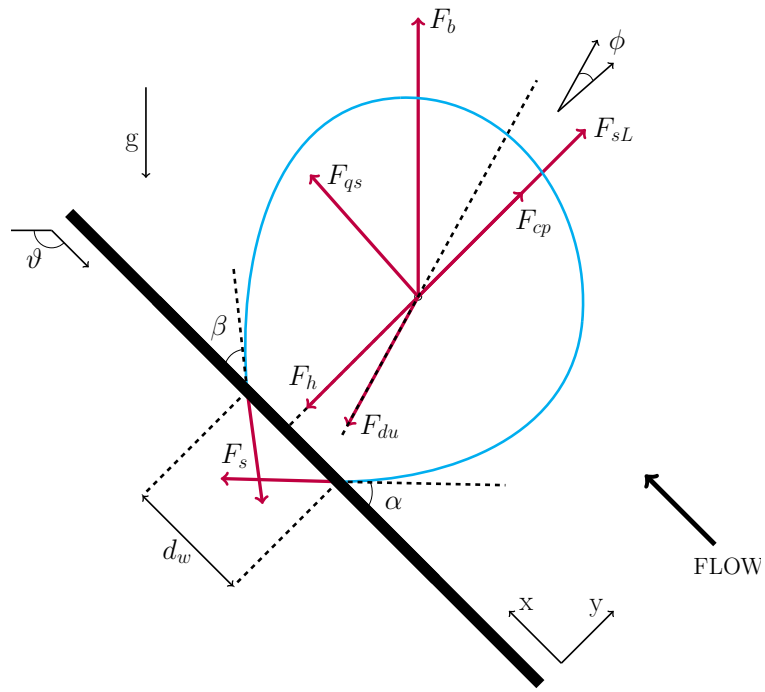


Figure 9.1. Illustration of the forces acting on a bubble in subcooled flow boiling. Figure provided by Rosemary Sugrue.

The bubble foot diameter on the heated surface was improved in the model by Sugrue et al. Klausner first suggested a constant value of 0.09 mm based on his experimental data using R113. Yun et al. [80] improved the model to account for a smaller bubble foot diameter that is present in high pressure and water and steam flows. The bubble departure diameter is strongly dependent on the bubble contact diameter, so Yun defined d_w to be a fraction of the bubble departure diameter that was given by (9.6). After further investigation by Sugrue, the bubble foot diameter was modified as that shown in (9.5).

$$\frac{d_w}{D_d} = \frac{1}{15} \tag{9.6}$$

For the turbulent flow conditions investigated, the dominant component of the unsteady drag is the quasi-steady drag force [87]. The expression was developed assuming an unbounded uniform flow over a spherical bubble and is shown in (9.7). The kinematic viscosity (ν_l) is calculated as μ_l/ρ_l. The constant n is set equal to 0.65. The instantaneous bubble radius is given by R.

$$F_{qs} = 6\pi\nu_l\rho_l UR\left[\frac{2}{3} + \left[\left(\frac{12}{\mathrm{Re_b}}\right)^n + 0.796^n\right]^{-1/n}\right] \tag{9.7}$$

The flow velocity (U) is shown in (9.8) and is taken to be the fluid velocity at the center of the bubble. Therefore, U is given as a function of the distance (y) from the wall. The time-averaged velocity profile is used because the evaluation is being completed for the mean bubble departure diameter. The time-averaged velocity near the wall is assumed to follow the turbulent single-phase relations given by Hinze [104]. The calculation of the bubble Reynolds number ($\mathrm{Re_b}$) employs this flow velocity and is calculated as shown in (6.17). The constants κ, χ, and c are set as 0.4, 11, and 7.4 respectively.

$$\frac{U(y)}{u^*} = \frac{1}{\kappa}\ln\left(1+\kappa y^+\right) + c\left[1-\exp\left(-\frac{y^+}{\chi}\right) - \frac{y^+}{\chi}\exp\left(-0.33y^+\right)\right] \tag{9.8}$$

The buoyancy force may act in either or both the x- and y-directions depending on the angle of the test-section and is shown in (9.9).

$$F_b = \frac{4}{3}\pi R^3\left(\rho_l - \rho_v\right)g \tag{9.9}$$

The hydrodynamic force is estimated considering the bubble is a sphere in an unbounded flow field [87] and is shown in (9.10). Due to the symmetry over the majority of the bubble surface, the hydrodynamic force is defined by the pressure on the top of the bubble over an area $\pi d_w^2/4$.

$$F_h = \frac{9}{8}\rho_l U^2 \frac{\pi d_w^2}{4} \tag{9.10}$$

The contact pressure force, shown in (9.11), is due to the pressure difference between the inside and outside of the bubble at the reference point over the contact area. It is dependent on the radius of curvature, and r_r is the radius of curvature of the bubble at the reference point on the surface. An estimation of $r_r = 5R$ proposed by Klausner [87] is used.

$$F_{cp} = \frac{\pi d_w^2}{4}\frac{2\sigma}{r_r} \tag{9.11}$$

Since there is not an expression for the shear lift force on a bubble attached to a wall, the shear lift force is formulated using an expression for the shear lift on a spherical bubble in an unbounded flow field [105] and is shown in (9.12). This was combined with the results for the shear lift force on a bubble in the inviscid flow limit with small shear rate [106] and interpolated to estimate the shear lift force over a large range of Reynolds numbers. The dimensionless shear rate (G_s) is given in (9.13).

$$F_{sL} = \frac{1}{2}\rho_l U^2 \pi R^2\left[3.877 G_s^{1/2}\left[\mathrm{Re}_b^{-2} + \left(0.344 G_s^{1/2}\right)^4\right]^{1/4}\right] \tag{9.12}$$

$$G_s = \left|\frac{dU}{dy}\right|\frac{R}{U} \tag{9.13}$$

The unsteady force due to asymmetrical bubble growth is caused by the formation of the bubble on the heated surface and its distorted shape due to the flow. The calculation is dependent on the inclination angle for the x- and y-components and is shown in (9.14).

$$F_{du} = -\rho_l \pi R^2\left(R\ddot{R} + \frac{3}{2}\dot{R}^2\right) \tag{9.14}$$

The force is also dependent on the growth rate of the bubble. The original Klausner model used an expression developed to predict the time rate of change for the radius of a spherical growing vapor bubble at a wall that was developed by Mikic et al. [107] and is shown in (9.15). The definitions for t^+, A, and B are shown in (9.16), (9.17), and (9.18) respectively. The unsteady force is also dependent on the first ($\dot{R}$) and second ($\ddot{R}$) time derivatives of the growth rate.

$$R(t) = \frac{2}{3}\frac{B^2}{A}\left[\left(t^+ + 1\right)^{3/2} - \left(t^+\right)^{3/2} - 1\right] \tag{9.15}$$

$$t^+ = \frac{A^2 t}{B} \tag{9.16}$$

$$A = \left[\frac{\pi}{7}\frac{h_{fg}\rho_v \Delta T_{sup}}{\rho_l T_{sat}}\right]^{1/2} \tag{9.17}$$

$$B = \left[\frac{12}{\pi}\eta_l\right]^{1/2} \frac{\Delta T_{sup} c_{pl}\rho_l}{h_{fg}\rho_v} \tag{9.18}$$

The improved bubble departure model by Sugrue employs the bubble growth model by Zuber [79], which is the same bubble growth model used by Yun [80] without the subcooling component and is shown in (9.19). The asphericity of the bubble is captured by using the constant b = 1.56. To be consistent with the development of the model by Sugrue et al., the wall superheat used to calculate the Ja number in the bubble growth model is held constant at 3 K for all test cases. The dominant forces are typically the shear lift, buoyancy, and surface tension forces, so the constant wall superheat assumption is acceptable in most cases.

$$R(t) = \frac{2b}{\sqrt{\pi}}\mathrm{Ja}\sqrt{\eta_l t} \tag{9.19}$$

9.2 Bubble Lift-off Diameter

When a bubble is sliding, the forces acting on the bubble are different from the model by Sugrue et al. [23] 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 is essentially zero. Additionally, it has been shown that the inclination angle goes to zero when the bubble slides, so the unsteady growth force is normal to the flow direction. The competing forces to predict the lift-off diameter were formulated to depend on the orientation of the heater [15, 27] and is shown in (9.20) and illustrated in Figure 9.2.

$$\Sigma F_y = F_{du} + F_{sL}^* - F_b \cos\vartheta \tag{9.20}$$

The shear lift force (F_{sL}^*) 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 of the bulk flow. 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 [43] also showed that the bubble is governed by the growth force and shear lift force at the instant of lift-off. The lift-off diameter is provided by the model when the summation of the forces is greater than zero.

9.3 Bubble Departure Frequency

A study was completed of available bubble departure frequency models to determine the most appropriate for the new boiling model. The more traditional models are highly dependent on the bubble departure diameter, such as Cole [39], Ivey [108], Stephan [109], and Jakob [110].

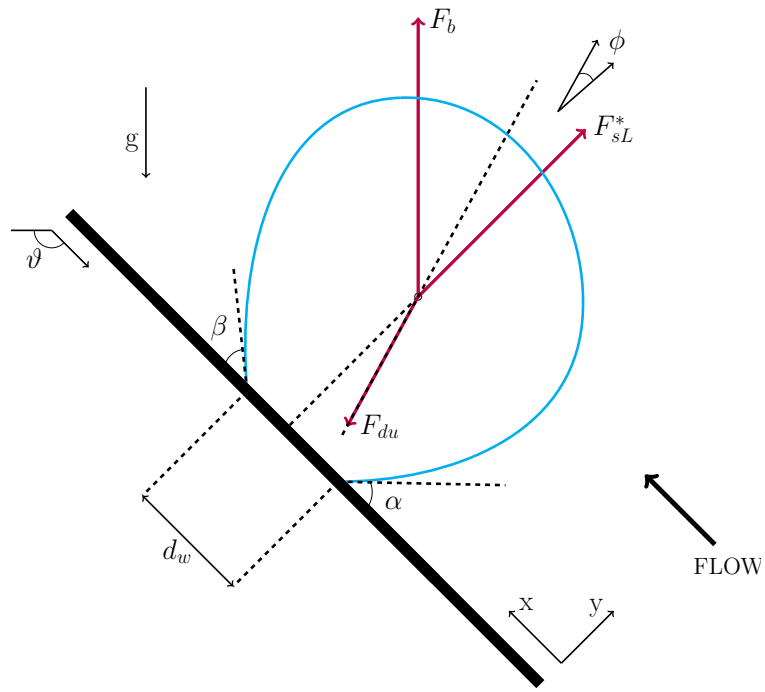


Figure 9.2. Illustration of the forces acting on a bubble in subcooled flow boiling.

The Cole bubble departure frequency correlation was derived assuming a balance between buoyancy and drag, with the drag coefficient constant, for pool nucleate boiling and is shown in (3.30). This is the most commonly employed frequency model in CFD simulations. The frequency model by Ivey was developed to be used for coalesced bubbles and is shown in (9.21).

$$f = 0.9\sqrt{\frac{g}{D_d}} \tag{9.21}$$

The frequency model by Stephan includes the effect of surface tension and was developed for pool boiling and is shown in (9.22). It can also be used for low-pressure subcooled flow boiling cases. Jakob's frequency model, shown in (9.23), was derived for pool boiling assuming that the wait time was equivalent to the growth time.

$$f = \frac{1}{\pi}\sqrt{\frac{g}{2}\left(D_d + \frac{4\sigma}{\rho_l g D_d}\right)} \tag{9.22}$$

$$f = 0.59\left[\frac{\sigma g\left(\rho_l - \rho_g\right)}{\rho_l^2 D_d^2}\right]^{0.25} \tag{9.23}$$

The more recent mechanistic approach to model the bubble departure frequency employs the bubble ebullition cycle where $f = 1/\left(t_w + t_g\right)$. Models that employ this method include Han and Griffith [111], Basu [92], and Yeoh and Tu [43]. Han and Griffith analytically solve for the waiting time, as shown in (9.24), and then assume that $t_w = 3t_g$ to solve for the frequency. An approximation for the minimum critical cavity radius that is used in this model is shown in (9.25).

$$t_w = \frac{9}{4\pi\alpha}\left[\frac{(T_w - T_{\inf})\,R_{c,approx}}{T_w - T_{sat}\left[1 + \left(2\sigma/R_{c,approx}\rho_g h_{fg}\right)\right]}\right]^2 \tag{9.24}$$

$$R_{c,approx} = \frac{2\sigma T_{sat}}{\rho_g h_{fg}\Delta T_{sup}} \tag{9.25}$$

The correlation developed by Basu calculates both the wait time (9.26) and and growth time (9.27) to determine the frequency [40]. The subcooling Jakob number is calculated as shown (9.28).

$$t_w = 139.1\left(\Delta T_{sup}^{-4.1}\right) \tag{9.26}$$

$$\frac{D_d^2}{\eta_l \mathrm{Jat}_g} = 45\exp\left(-0.02\mathrm{Ja}_{sub}\right) \tag{9.27}$$

$$\mathrm{Ja}_{sub} = \frac{\rho_l c_{pl}\Delta T_{sub}}{\rho_v h_{fg}} \tag{9.28}$$

The frequency model developed by Yeoh and Tu is similar to the derivation by Han and Griffiths, but it adds an additional parameter that is dependent on the bubble contact angle. The wait time calculation is given in (9.29) where C_1 and C_2 are shown in (9.30) and (9.31) respectively. The recommendation for the calculation for the growth time is to use an appropriate bubble growth model for the conditions being simulated.

$$t_w = \frac{1}{\pi\eta_l}\left[\frac{(T_w - T_l)\,R_c C_1}{(T_w - T_{sat}) - 2\sigma T_{sat}/C_2 R_c \rho_g h_{fg}}\right]^2 \tag{9.29}$$

$$C_1 = (1 + \cos\theta)\,/\sin\theta \tag{9.30}$$

$$C_2 = 1/\sin\theta \tag{9.31}$$

A new bubble departure frequency was developed based on the Yeoh and Tu [43] model. The growth time is calculated using the same bubble growth equation that is used in the force-balance bubble departure model which is Zuber's model [79]. When the bubble departure diameter is known through the force-balance prediction, the growth time is calculated as shown in (5.13).

The wait time is calculated using the fractal model developed by Yeoh and Tu [43] given in (9.29). The new frequency model uses the approximate value for the minimum critical radius given in (9.25). Although this new frequency model has the advantage of being formulated mechanistically, it was not implemented in the new boiling model simulations because calculations involving the factor fN'' were not accurate since the nucleation site density models were not developed with the frequency model. Additionally, implementation of more detailed and mechanistic frequency models into the CFD software caused great difficulty in obtaining converged results. Therefore, the Cole (1960) bubble departure frequency model was implemented in the new boiling model because it is the most numerically stable frequency model, and also predicts values that compare reasonably well to experimental data.

CHAPTER THREE: TESTING AND VALIDATION OF NEW MODEL

10 Model Validation

The wall temperature predictions by the new boiling model (Gilman 2014) described in Chapter 2 is compared to both experimental data and the standard wall boiling model predictions. The standard model used for comparison employs the heat partitioning approach by Kurul and Podowski (1990) [53] and the closure models as described in Section 3.3. The new model has improved description of the physical phenomena that occur at the wall during subcooled boiling so that it can more accurately predict the wall surface temperature. Experimental data is used to compare the wall temperature predictions of both models at various pressures, mass fluxes, heat fluxes, and inlet subcoolings.

10.1 Comparison with MIT Flow Boiling Facility Data

The MIT Flow Boiling Facility described in Section 2.4 collected high-quality temperature data of the heated surface using IR thermography. The temperature of the heated surface was averaged over an area of ∼6 mm wide and ∼5 mm tall centered on the heater.

For each test case, the full geometry was built in the STAR-CCM+ CAD by extruding a shortened geometry in the direction of the flow. The original CAD is shown in Figure 10.1, with the length in the z-direction being 220 mm. The meshing models used included the prism layer mesher, surface remesher, trimmer, and extruder. The mesh was extruded in the negative z-direction using 200 layers, a stretching value of 1.0, and a distance of 965.2 mm. The final extruded geometry is shown in Figure 10.2.

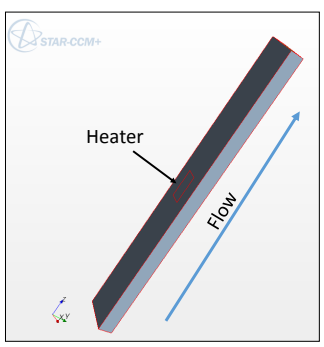


Figure 10.1. Original CAD geometry of the quartz cell region.

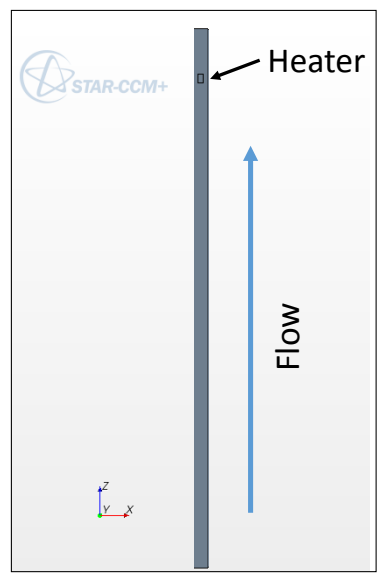


Figure 10.2. Full geometry (after extrusion) showing the vertical upflow position.

A constant velocity calculated from the experimental mass flux was imposed at the inlet of the full channel. The velocity profile at the quartz cell entrance, for the x-,y-, and z-directions, was exported and saved to be used as the velocity inlet boundary condition on a shortened geometry. A new geometry was then built that only included the quartz cell region to reduce the number of computational cells in the geometry. The inlet temperature, which defines the inlet subcooling, was imposed at the inlet boundary. Every test

case imposed a constant pressure outlet boundary condition. Also at the outlet, the backflow direction specification was set to boundary-normal, the pressure specification was set as environmental, and the volume fraction specification imposed was pure liquid.

The constant density EMP standard wall boiling models as described in Section 3.3 were implemented in all test cases. In the testing of the new wall boiling model, the mechanistic bubble departure diameter model was used in place of the Tolubinsky-Kostanchuk model. The advancing and receding contact angles used in the bubble departure models are those for water on ITO and are 100° and 25° respectively [44]. Also, in the new model, the rough wall model was implemented instead of the smooth wall model to properly capture the increased turbulence near the wall caused by the presence of the bubbles. The properties of the heater used in the calculation of the quenching component were 6800 kg/m^3 for the density and 500 J/kg-K for the specific heat [112]. All thermophysical properties of the fluids implemented in the test cases were gathered from the National Institute of Standards and Technology (NIST) [113].

10.1.1 Wall Temperature Predictions at 1.05 Bar

The new wall boiling model was tested at atmospheric pressure for mass fluxes of 500, 1000, and 1250 kg/m^2-s, which corresponds to Reynolds numbers of 2.5x10^4, 5.0x10^4, and 6.3x10^4. All cases were tested with an inlet subcooling of 10 K.

Due to the instability of the simulations while using the Hibiki-Ishii nucleation site density model that is available in the STAR-CCM+ software, an exponential curve was fitted for the given pressure conditions and bubble contact angle and implemented in both the Gilman and Kurul-Podowski models. The static contact angle used for water on the ITO heater was 100° [45]. The implementation of the nucleation site density through a field function allowed for the user to impose a relaxation factor to aid in convergence. The nucleation site density curve implemented in the test cases run at a pressure of 1.05 bar is shown in (10.1). The nucleation site density calculation was relaxed as shown in (10.2), where N_1'' represents the nucleation site density based on the current iteration wall superheat and N_0'' represents the nucleation site density based on the previous iteration wall superheat. The relaxation factor (RF) was typically set to a value of 0.10 and was adjusted to aid in convergence if required for the simulation.

$$N'' = -19906.7\left(1 - e^{0.20332\Delta T_{sup}}\right) \tag{10.1}$$

$$N_1'' = RF\left(N_1'' - N_0''\right) + N_0'' \tag{10.2}$$

A monitor was used in the STAR-CCM+ simulation file to save the nucleation site density value from the previous iteration (N_0''). The new field mean monitor was was set to trigger at each iteration. A sliding window was implemented with the sliding sample window size equal to one. The field function "Nucleation Site Number Density" was selected for the heater region. This field monitor produced a new field function titled "Mean of Nucleation Site Number Density" which contained the nucleation site density value from the previous iteration.

To ensure that the velocity profile near the wall was accurately modeled, the wall y$^+$ values were inspected to confirm they were in the range of 30-50. Since the velocity profile varies for each imposed mass flux, the cell sizes (in particular the prism layer) were defined for each mass flux to satisfy the criteria required for proper use of the turbulence models.

Mass flux of 500 kg/m^2-s

For the test case with a mass flux of 500 kg/m^2-s (0.52 m/s), the cell sizes used for meshing the quartz region geometry are shown in Table 10.1. In a region of 50 mm in length, and centered over the heater area, a volumetric mesh control was implemented to reduce the cell size by half in the y-direction (the direction

normal to the wall) since this is the region where boiling will occur. A cross-section of the mesh at the center of the heater is shown in Figure 10.3.

Table 10.1. Meshing values used for the MIT Flow Boiling Facility test cases that have a mass flux of 500 kg/m^2-s.

Parameter	Value
Base size	0.9 mm
Maximum cell size (relative to base)	100%
Number of prism layers	1
Prism layer thickness	0.9 mm
Surface growth rate	1.3
Template growth rate	Fast

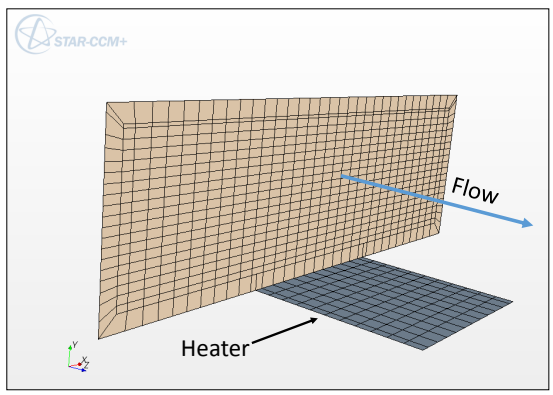


Figure 10.3. Cross-section of the volume mesh at the center of the heater for the MIT Flow Boiling Facility test cases that have a mass flux of 500 kg/m^2-s.

The value for the wall y$^+$ for the converged velocity solution for the test cases with a mass flux of 500 kg/m^2-s is ~43 and is shown in Figure 10.4. This value for the wall y$^+$ ensures that first cell on the wall boundary is the proper size for modeling the velocity profile using the standard k-ε turbulence model.

The experimental data for a mass flux of 500 kg/m^2-s and an inlet subcooling of 10K was taken for heat fluxes of 100-1000 kW/m^2 and the heater temperature was averaged over the center of the heater. The standard set of models (Kurul-Podowski) were implemented and the wall temperature prediction for heat fluxes of 100-900 kW/m^2 are shown in Figure 10.5. The simulation using the standard wall boiling models did not converge at a heat flux of 1000 kW/m^2. The new model (Gilman) was implemented for heat fluxes of 100-1000 kW/m^2 and the averaged heater temperatures are also compared to the experimental data in Figure 10.5.

The results shown in Figure 10.5 illustrate that both the Kurul-Podowski and Gilman model predict the wall temperatures fairly accurately for the test conditions at 1.05 bar. Both models overpredict the temperature (~2-7 K) at lower heat fluxes (below 500 kW/m^2), with the Gilman model being slightly closer to the experimental data. The overprediction of the wall temperature at low heat fluxes may be caused by the shape of the Hibiki-Ishii active nucleation site density curve because it provides very low values of active sites at lower wall superheats. When there are less active nucleation sites, the heat is removed less efficiently from the surface, resulting in higher wall temperatures. Above 500 kW/m^2, the Gilman model remains within the experimental error.

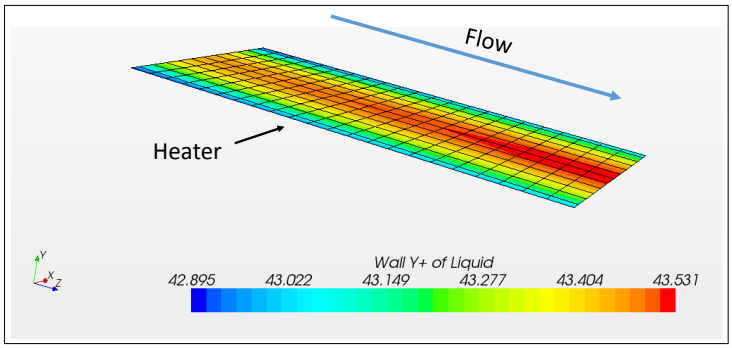


Figure 10.4. Wall y^+ values on the heater region for the MIT Flow Boiling test cases that have a mass flux of 500 kg/m^2-s.

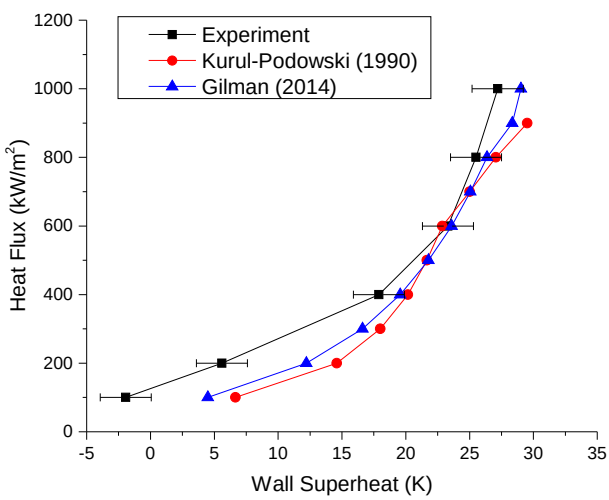


Figure 10.5. Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 500 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 1.05 bar.

Dry Surface Area

The dry surface area was calculated as described in Section 7 for all heat fluxes in the test case with a mass flux of 500 kg/m^2-s and inlet subcooling of 10 K. The percent of heater surface dry area is shown in Table 10.2 and illustrates the steady increase in dry surface area as the heat flux increases. The calculation of the dry area is also highly dependent on the bubble departure size.

Table 10.2. The fraction of dry surface area for a mass flux of 500 kg/m^2-s, inlet subcooling of 10 K, and pressure of 1.05 bar.

Heat Flux [kW/m^2]	Dry Surface
100	0.005%
200	0.03%
300	0.07%
400	0.09%
500	0.24%
600	0.34%
700	0.44%
800	0.56%
900	0.81%
1000	1.10%

Contribution of Components in New Heat Partitioning Model

The significance of each partitioned component of the model in the overall heat removal from the surface was investigated by calculating the contribution of each component at all heat fluxes used to produce the boiling curve and the results are shown in Figure 10.6. These results show that the forced convection and sliding conduction components compete for the greatest effect on heat removal. It is important to note here that the sliding conduction component includes all heat transfer associated with disturbance of the thermal boundary layer in addition to that caused by sliding bubbles.

The forced convection component starts as the largest component and then decreases to about zero at the highest heat flux. This is expected since the bubble influence on the heater increases as the active nucleation site density increases. Since the bulk liquid remains subcooled at all heat fluxes, the transient (sliding) conduction component becomes the highest mode of heat transfer as the heat flux increases. The evaporation and quenching components increase steadily as the active nucleation site density increases with the wall temperature. At a heat flux of 1000 kW/m^2, the evaporation component makes removes $\sim$8% of the total heat flux. The evaporation component is highly dependent on the bubble departure diameter prediction, so the modification of the mechanistic bubble departure diameter model used in the study to include a dependence on the wall superheat could result in the components' contributions shifting as the bubble sizes change with the wall superheat.

Mass flux of 1000 kg/m^2-s

The cell sizes used for meshing the quartz region geometry are shown in Table 10.3 for the test cases that have a mass flux of 1000 kg/m^2-s (1.05 m/s). A volumetric control was used in the heater area to reduce the cell size by half in the y-direction. A cross-section of the mesh at the center of the heater is shown in Figure 10.7. The value for the wall y$^+$ in the converged velocity solution is $\sim$40 and is shown in Figure 10.8. This wall y$^+$ value ensures that the prism layer is the proper size.

The experimental data for a mass flux of 1000 kg/m^2-s was taken for heat fluxes of 200-1200 kW/m^2. The

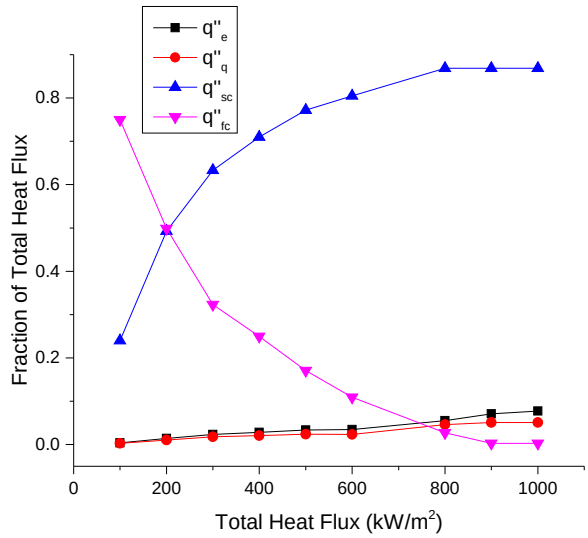


Figure 10.6. The fraction of the total heat flux removed by each component in the new model's heat partitioning model at a mass flux of 500 kg/m^2-s, 10 K subcooling, and a pressure of 1.05 bar.

Table 10.3. Meshing values used for the MIT Flow Boiling Facility test cases that have a mass flux of 1000 kg/m^2-s.

Parameter	Value
Base size	0.9 mm
Maximum cell size (relative to base)	100%
Number of prism layers	2
Prism layer thickness	0.9 mm
Thickness of Near Wall Prism Layer	0.6 mm
Surface growth rate	1.3
Template growth rate	Fast

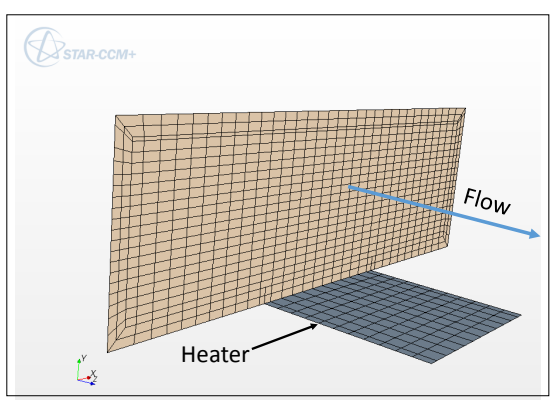


Figure 10.7. Cross-section of the volume mesh at the center of the heater for the MIT Flow Boiling Facility test cases that have a mass flux of 1000 kg/m^2-s.

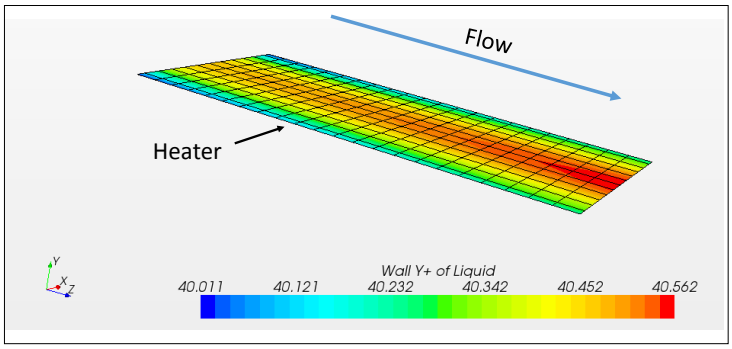


Figure 10.8. Wall y^+ values on the heater region for the MIT Flow Boiling test cases that have a mass flux of 1000 kg/m^2-s.

average heater temperatures of the experiment are shown in Figure 10.9. The standard wall boiling models were implemented to calculate the wall temperature for heat fluxes of 200-900 kW/m^2 and the averaged heater temperatures are also shown in Figure 10.9. The test cases using the standard wall boiling models did not converge for heat fluxes greater than 900 kW/m^2. The new model was implemented and tested using heat fluxes of 200-1200 kW/m^2. The averaged heater temperatures are compared to the experimental data and the standard model predictions in Figure 10.9.

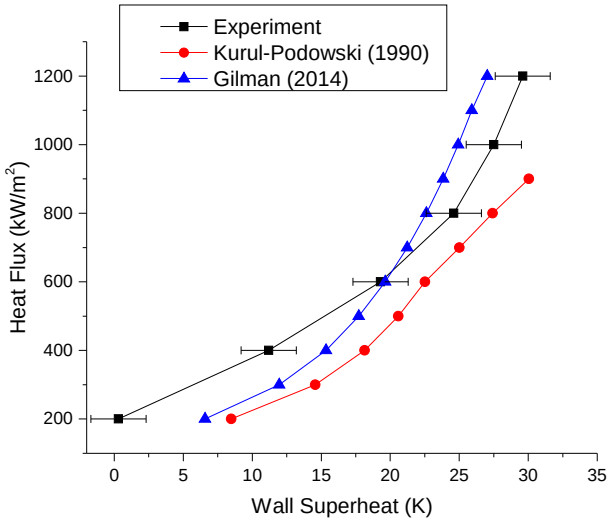


Figure 10.9. Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1000 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 1.05 bar.

The results shown in Figure 10.9 for a mass flux of 1000 kg/m^2-s have a similar trend to the temperature predictions for a mass flux of 500 kg/m^2-s. At lower heat fluxes, below 500 kW/m^2, both models over-predict the wall temperature by ~2-7 K. This is again likely caused by the active nucleation site density predictions of the Hibiki-Ishii model. As the heat flux increases, the Gilman model is closer to the experimental results and the Kurul-Podowski model begins to deviate from the shape of the boiling curve until it no longer converges to a solution at a heat flux of 1000 kW/m^2.

Mass flux of 1250 kg/m^2-s

The cell sizes used for meshing the quartz region geometry for the test cases that have a mass flux of 1250 kg/m^2-s (1.29 m/s) are shown in Table 10.4. A volumetric mesh control was also used to reduce the cell size by half in the y-direction in the area around the heater. A cross-section of the mesh at the center of the heater is shown in Figure 10.10. The value for the wall y$^+$ for the converged velocity solution is ∼43 is shown in Figure 10.11 to illustrate that the prism layer is the correct size.

Table 10.4. Meshing values used for the MIT Flow Boiling Facility test cases that have a mass flux of 1250 kg/m^2-s.

Parameter	Value
Base size	0.9 mm
Maximum cell size (relative to base)	100%
Number of prism layers	2
Prism layer thickness	0.8 mm
Thickness of Near Wall Prism Layer	0.45 mm
Surface growth rate	1.3
Template growth rate	Fast

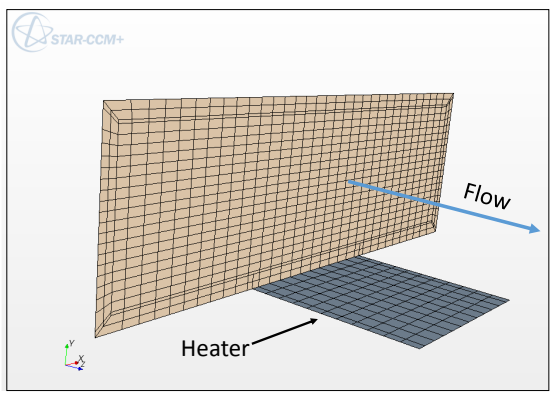


Figure 10.10. Cross-section of the volume mesh at the center of the heater for the MIT Flow Boiling Facility test cases that have a mass flux of 1250 kg/m^2-s.

The experimental data for a mass flux of 1250 kg/m^2-s was also taken for heat fluxes ranging from 200-1200 kW/m^2 at a pressure of 1.05 bar. The standard wall boiling models were implemented to calculate the wall temperature for heat fluxes of 200-1000 kW/m^2 and the averaged temperature of the heater is shown in Figure 10.12. The simulations using the standard wall boiling models did not converge for heat fluxes greater than 1000 kW/m^2. The new model was also implemented and tested with heat fluxes of 200-1200 kW/m^2. The averaged heater temperatures are shown in Figure 10.12 and compared to the experimental data and the standard model predictions.

The results for the mass flux of 1250 kg/m^2-s shown in Figure 10.12 continue the temperature trends as described for the mass fluxes of 500 and 1000 kg/m^2-s. At lower heat fluxes (below 600 kW/m^2), the temperatures are slightly overpredicted by both models (∼1-6 K). At higher heat fluxes, both models begin to underpredict the wall temperatures. This may be caused by the inability of the Hibiki-Ishii nucleation site density model to account for the effect of the increased mass flux that reduces the number of active nucleation sites. The reduction of active nucleation sites would result in higher wall temperature predictions since less boiling and bubble disturbance of the thermal boundary layer would occur. The Gilman model

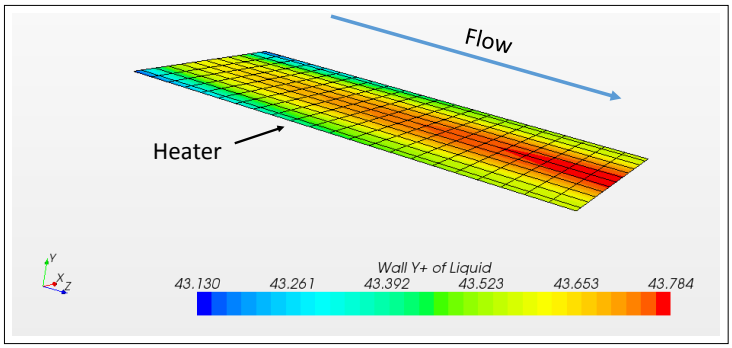


Figure 10.11. Wall y^+ values on the heater region for the MIT Flow Boiling test case with a mass flux of 1250 kg/m^2-s.

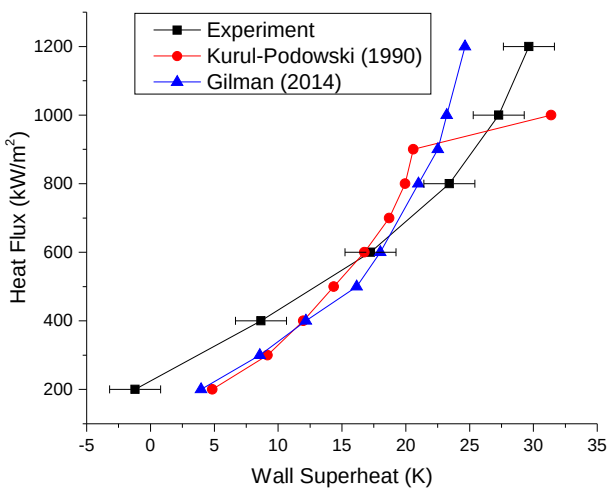


Figure 10.12. Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1250 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 1.05 bar.

more closely resembles the experimental boiling curve and the Kurul-Podowski model does not converge at the highest heat fluxes.

10.1.2 Wall Temperature Predictions at 1.5 Bar

The new wall boiling model was tested at a pressure of 1.5 bar for mass fluxes of 500, 1000, and 1250 kg/m^2-s. The same geometry setup and mesh size for each mass flux as described in Section 10.1.1 was used. All cases were tested with an inlet subcooling of 10 K. A new nucleation site density exponential fit curve was made using the Hibiki-Ishii correlation for a pressure of 1.5 bar and is shown in (10.3). The same relaxation method as shown in (10.2) was implemented to aid in convergence.

$$N'' = -13558\left(1 - e^{0.24295\Delta T_{sup}}\right) \tag{10.3}$$

The average heater temperatures for the new (Gilman 2014) and the standard (Kurul-Podowski 1990) wall boiling models versus the experimental data for a mass flux of 500 kg/m^2-s and heat fluxes of 100-1400 kW/m^2 are shown in Figure 10.13. The standard model did not converge for heat fluxes greater than 1000 kW/m^2.

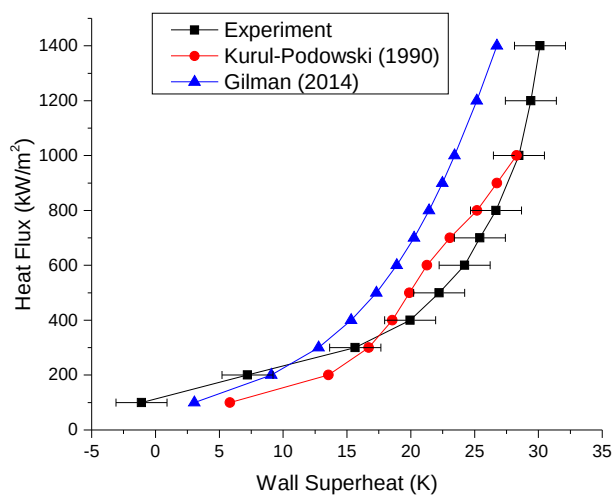


Figure 10.13. Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 500 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 1.5 bar.

The significance of each partitioned component of the model in the overall heat removal from the surface was also investigated for a pressure of 1.5 bar and the results are shown in Figure 10.14. These results again show that the forced convection and sliding convection components compete for the largest component for heat removal. The forced convection component starts as the largest component and then decreases quickly as the heat flux increases. Since the bulk liquid remains subcooled at all heat fluxes, the transient conduction component becomes the highest mode of heat transfer as the heat flux increases. The evaporation and quenching components increase steadily as the active nucleation site density increases with the wall temperature and have a smaller contribution since the bubble departure diameter is smaller at higher pressures.

The new and the standard wall boiling models temperature predictions were compared to the experimental data for a mass flux of 1000 kg/m^2-s and heat fluxes of 100-1400 kW/m^2 in Figure 10.15. The standard model was not able to converge for heat fluxes greater than 900 kW/m^2.

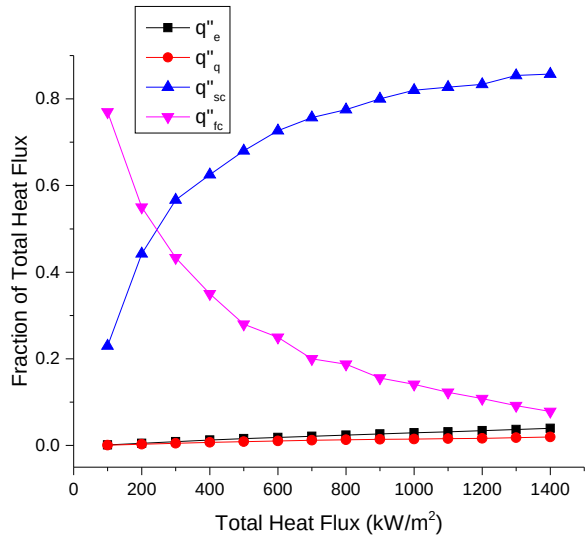


Figure 10.14. The fraction of the total heat flux removed by each component in the new model's heat partitioning model at a mass flux of 500 kg/m^2-s, 10 K subcooling, and a pressure of 1.5 bar.

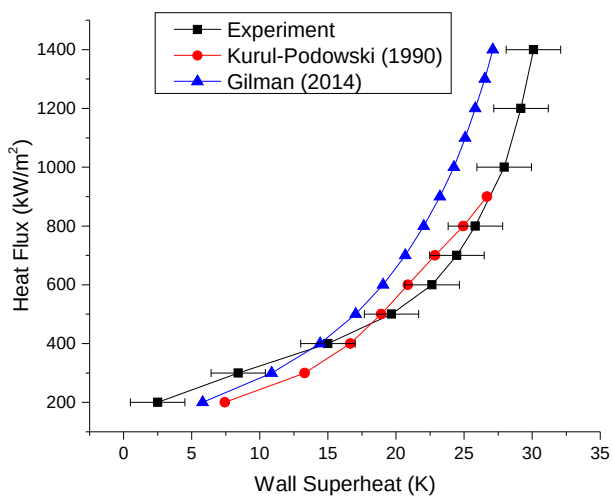


Figure 10.15. Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1000 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 1.5 bar.

The temperature predictions for the new and standard wall boiling models versus the experimental data for a mass flux of 1250 kg/m^2-s and heat fluxes of 200-1600 kW/m^2 are shown in Figure 10.16. The standard model was not able to converge for heat fluxes greater than 1100 kW/m^2.

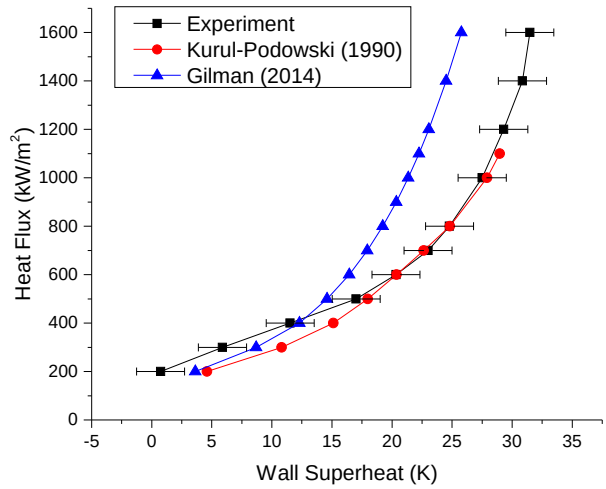


Figure 10.16. Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1250 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 1.5 bar.

The results for the three simulations shown in Figures 10.13, 10.15, and 10.16 display similar trends for the wall temperature predictions. At lower heat fluxes both models predict the wall temperature fairly accurately. The Gilman model is slightly closer to the experimental data at these low heat fluxes. The Gilman model then begins to underpredict the temperatures as the heat flux increases, while the Kurul-Podowski model tends to be closer to the experimental data (but does not converge at the highest heat fluxes). The deviation of the Gilman model from the experimental data may be caused by the overprediction of the bubble lift-off diameter. As the bubble lift-off diameter prediction becomes larger, the heat transfer coefficient of the sliding conduction term, the evaporation term, and the quenching term increases, which results in lower wall temperatures.

10.1.3 Wall Temperature Predictions at 2 Bar

The new wall boiling model was tested at a pressure of 2.0 bar for mass fluxes of 500, 1000, and 1250 kg/m^2-s. The same geometry setup and mesh size for each mass flux as described in Section 10.1.1 was used. All cases were tested with inlet subcoolings of 5, 10, and 15 K. A new nucleation site density exponential fit curve was made using the Hibiki-Ishii correlation for a pressure of 2.0 bar and is shown in (10.4). The same relaxation method as shown in (10.2) was implemented to aid in convergence.

$$N'' = -6091\left(1 - e^{0.3016\Delta T_{sup}}\right) \tag{10.4}$$

The average heater temperature predictions using the new (Gilman 2014) and standard (Kurul-Podowski 1990) wall boiling models were compared to the experimental data for a mass flux of 500 kg/m^2-s and heat fluxes of 100-1400 kW/m^2. The results for inlet subcoolings of 5, 10, and 15 K are shown in Figures 10.17, 10.18, and 10.19 respectively. The standard model was not able to converge for heat fluxes greater than 800 kW/m^2 for an inlet subcooling of 5K, 1000 kW/m^2 for an inlet subcooling of 10K, and 1200 kW/m^2 for an inlet subcooling of 15K.

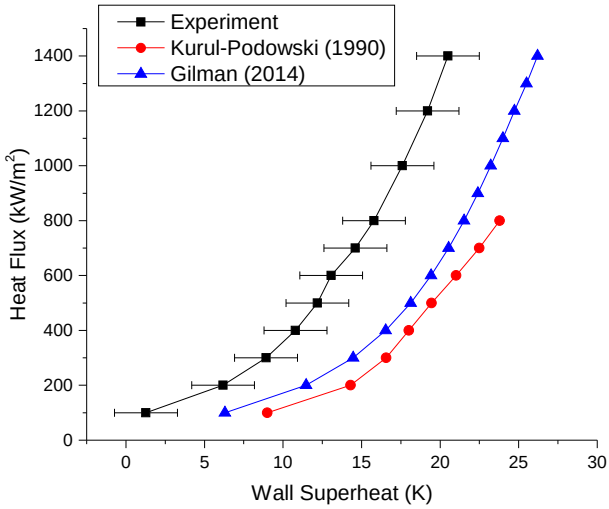


Figure 10.17. Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 500 kg/m^2-s, inlet subcooling of 5 K, and at a pressure of 2.0 bar.

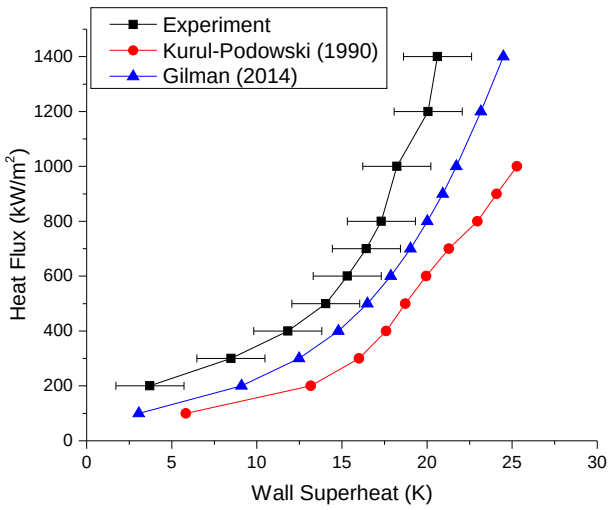


Figure 10.18. Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 500 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 2.0 bar.

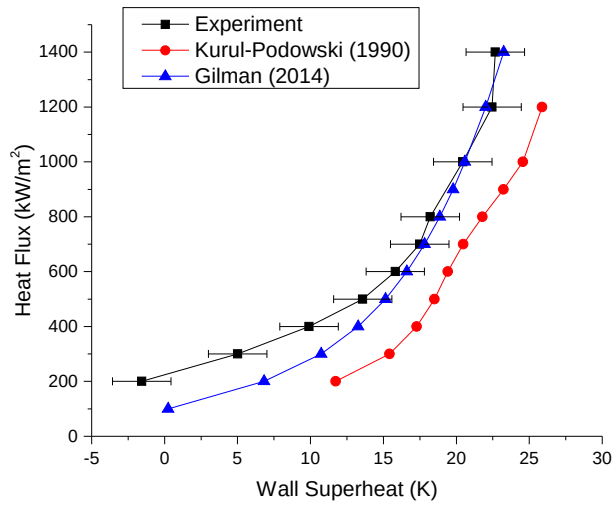


Figure 10.19. Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 500 kg/m^2-s, inlet subcooling of 15 K, and at a pressure of 2.0 bar.

The new and standard wall boiling models average heater temperature predictions were compared to the experimental data for a mass flux of 1000 kg/m^2-s and heat fluxes of 200-1400 kW/m^2. The results for inlet subcoolings of 5, 10, and 15 K are shown in Figures 10.20, 10.21, and 10.22 respectively. The standard model was not able to converge for heat fluxes greater than 600 kW/m^2 for an inlet subcooling of 5 K, 900 kW/m^2 for an inlet subcooling of 10 K, and 1200 kW/m^2 for an inlet subcooling of 15 K.

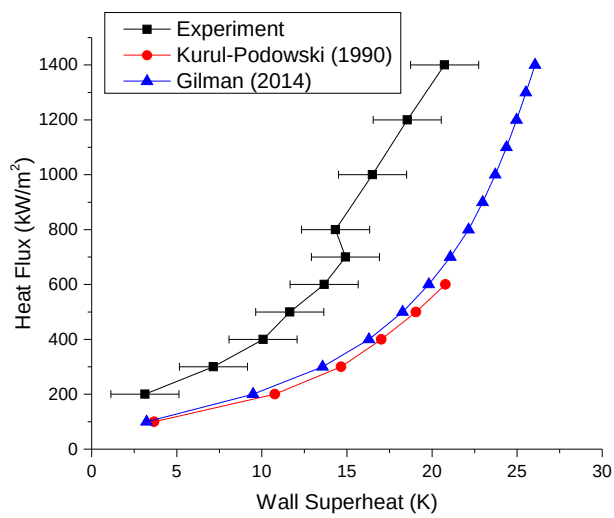


Figure 10.20. Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1000 kg/m^2-s, inlet subcooling of 5 K, and at a pressure of 2.0 bar.

The average heater temperature predictions for the new and standard wall boiling models were compared to the experimental data for a mass flux of 1250 kg/m^2-s and heat fluxes of 200-1600 kW/m^2. The results for inlet subcoolings of 5, 10, and 15K are shown in Figures 10.23, 10.24, and 10.25 respectively. The standard

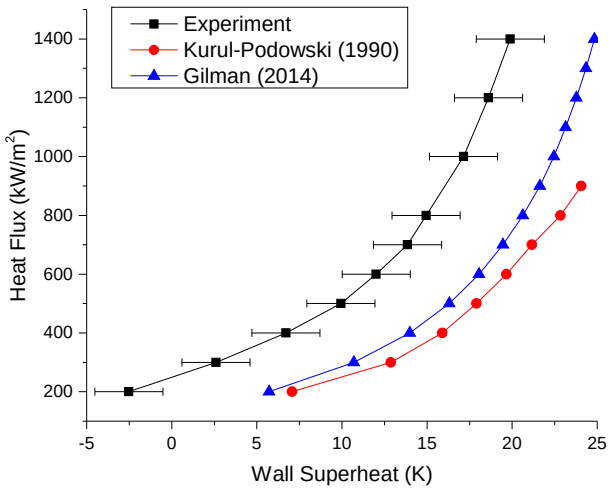


Figure 10.21. Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1000 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 2.0 bar.

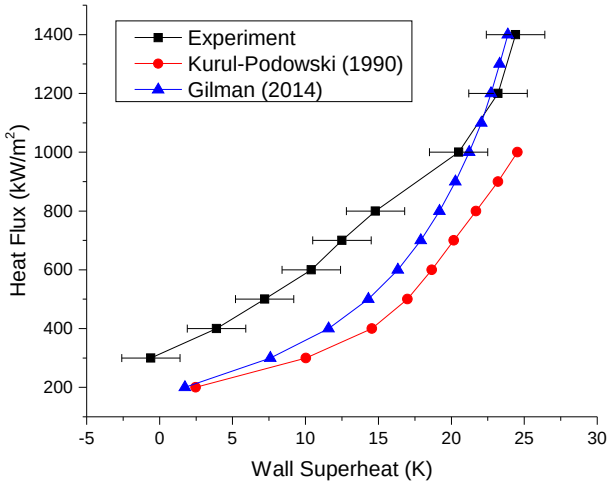


Figure 10.22. Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1000 kg/m^2-s, inlet subcooling of 15 K, and at a pressure of 2.0 bar.

model was not able to converge for heat fluxes greater than 800 kW/m^2 for an inlet subcooling of 5 K, 1300 kW/m^2 for an inlet subcooling of 10 K, and 1400 kW/m^2 for an inlet subcooling of 15 K.

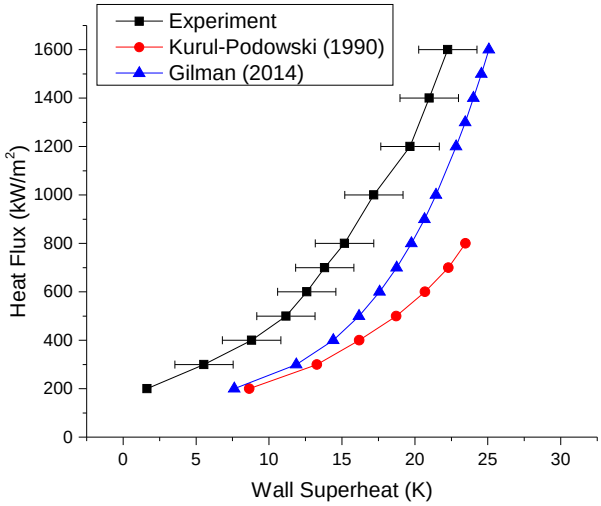


Figure 10.23. Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1250 kg/m^2-s, inlet subcooling of 5 K, and at a pressure of 2.0 bar.

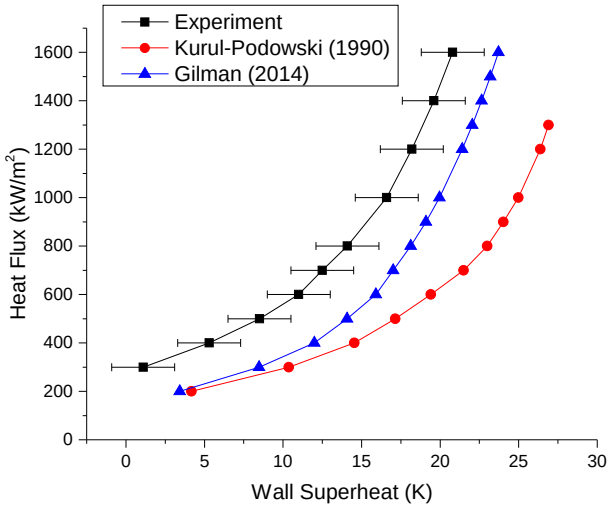


Figure 10.24. Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1250 kg/m^2-s, inlet subcooling of 10 K, and at a pressure of 2.0 bar.

Both the Gilman and Kurul-Podowski models overpredict the wall temperature at all heat fluxes for the test cases that have an inlet subcooling of 5 K (results shown in Figures 10.17, 10.20, and 10.23). The shape of the boiling curve predicted by the Gilman model is very similar to the experimental data, but the curve is shifted to higher wall temperatures (to the right) by about 5-8 K for the mass fluxes of 500 and 1000 kg/m^2-s. The Gilman model overpredicts the wall temperatures of the test case with a mass flux of 1250 kg/m^2-s by about 3-5 K. In all three cases for a subcooling of 5 K, the Kurul-Podowski model overpredicts the wall temperatures more than the Gilman model.

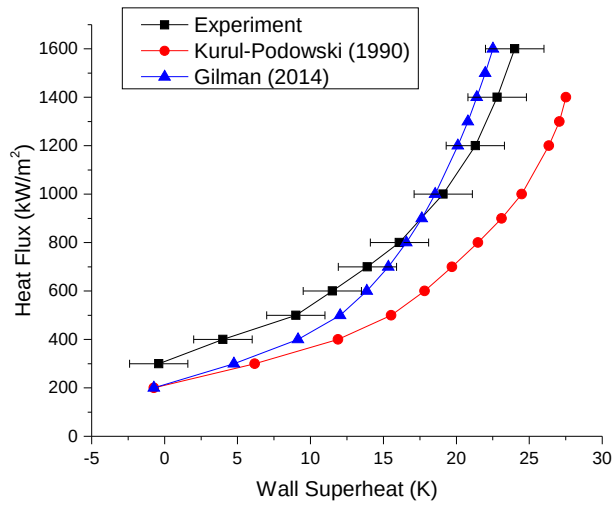


Figure 10.25. Average heater temperature predictions for the experiment, Kurul-Podowski model, and Gilman model for a mass flux of 1250 kg/m^2-s, inlet subcooling of 15 K, and at a pressure of 2.0 bar.

The test cases with an inlet subcooling of 10 K (results are shown in Figures 10.18, 10.21, and 10.24) also overpredict the wall temperatures for both models, but to a lesser extent as compared to the cases with an inlet subcooling of 5 K. The simulations run using the Gilman model with mass fluxes of 500 and 1250 kg/m^2-s overpredict the wall temperature by about 1-5 K. In both cases, the Kurul-Podowski model overpredicts the wall temperature more than the Gilman model and does not converge at higher heat fluxes. The Gilman model overpredicts the wall temperatures more for the mass flux of 1000 kg/m^2-s case than the other two cases. The Kurul-Podowski model predicts higher wall temperatures than the Gilman model for this test case as well.

In the results for an inlet subcooling of 15 K (shown in Figures 10.19, 10.22, and 10.25), the Gilman model overpredicts the wall temperature slightly at the low heat fluxes, and then matches the experimental data very closely as the heat flux increases above about 600kW/m^2. The Kurul-Podowski model overpredicts the wall temperatures more than the Gilman model and does not converge at the highest heat fluxes.

10.2 Comparison with High Pressure Rohsenow Data

Although numerous experimental data is available at low pressure for subcooled boiling with water, many experiments use refrigerants in place of water as the working fluid to simulate high pressure conditions. A specific refrigerant and test conditions are chosen to capture the relevant thermal-hydraulic conditions of interest. For example, the DEBORA tests [35] used Freon 12 because it is scaleable to typical PWR conditions. Other experiments that have used refrigerants to simulate high pressure water conditions include those by Hasan et al. [114] and Roy et al. [115].

In the testing of the new wall boiling model, a high pressure experimental test case using water as the working fluid was used since refrigerants have very different latent heat of evaporization, surface tension, and contact angle values. As a result, the Rohsenow et al. [116] data was used to compare the wall temperature predictions of the new wall boiling model because it included a detailed description of the experimental setup, employed water as the working fluid, and had wall temperature data at a pressure of 2000 psia (137.9 bar).

The experimental setup used a vertical, electrically heated, pure nickel test section 9.4 inches long. The inner and outer diameters of the tube were 0.1805 and 0.2101 inches respectively. The outer wall temperature

was measured in 7 locations (spaced 1.4 inches apart) and electrically insulated from the tube wall using a small sheet of mica 0.0015 inches thick. The locations of the measurements used in the data collection (thermocouples 2-6) are shown in Figure 10.26.

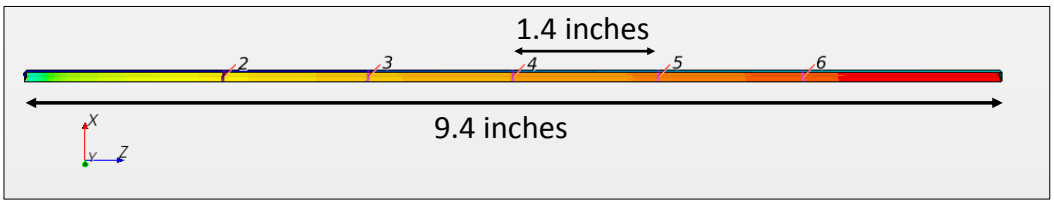


Figure 10.26. The locations of the outer wall temperature measurements 2, 3, 4, 5, and 6 for the Rohsenow test data.

The tube was modeled in STAR-CCM+ in 3-D using symmetry boundaries on a quarter of the tube geometry. The original CAD, shown in Figure 10.27, is a shortened length (0.01 m) of the full tube. The remaining length of the tube (0.22876 m) was extruded from the outlet to create cells with a higher aspect ratio in the direction of the flow, which are illustrated in Figure 10.28 near the outlet of the tube. The meshing values are given in Table 10.5 and a cross-section of the mesh is shown in Figure 10.29. The value of the wall y^+ on the wall was ∼30 and is shown in Figure 10.30.

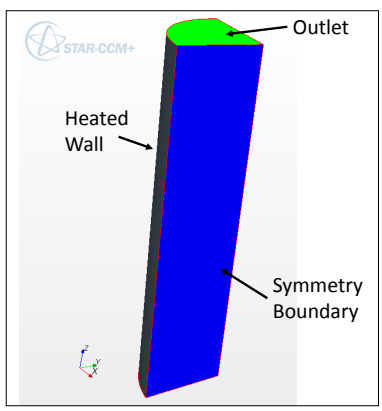


Figure 10.27. The Rohsenow test section original CAD geometry.

A new nucleation site density curve was developed from the Hibiki-Ishii nucleation side density model for a pressure of 2000 psi. The value used for the contact angle of water on Ni was 45° [117, 118]. Since the nucleation site density increases extremely quickly at low wall superheats at this high pressure, the nucleation site density curves were fitted over ranges of wall superheat. The fitted curves were completed for temperatures of 0 to 2 K, 2 to 5 K, and 5 to 9 K and shown in (10.5), (10.6), and (10.7) respectively. When the wall superheat was in the prescribed temperature range of a curve, the active nucleation site density value was predicting using that curve. The same relaxation method as shown in (10.2) was implemented to aid in convergence with the typical relaxation factor value being 0.01.

$$N'' = -63448.6\left(1 - e^{4.1275\Delta T_{sup}}\right) \tag{10.5}$$

$$N'' = -55254\left(1 - e^{4.16943\Delta T_{sup}}\right) \tag{10.6}$$

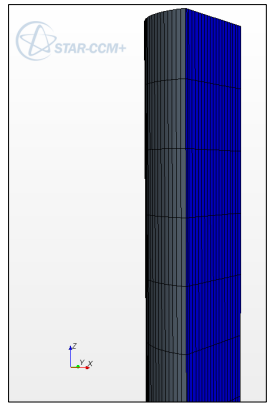


Figure 10.28. The extruded Rohsenow test case geometry near the outlet of the tube.

Table 10.5. Meshing values used for the Rohsenow test cases that have an inlet velocity of 3.048 m/s.

Parameter	Value
Base size	0.12 mm
Maximum cell size (relative to base)	100%
Number of prism layers	2
Prism layer thickness (absolute size)	0.12 mm
Prism layer stretching	1.1
Template growth rate	Fast
Extrusion layers	150
Stretching	1.0

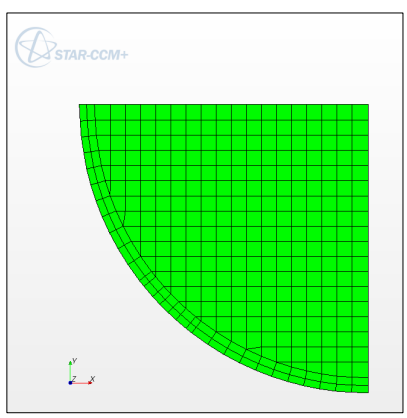


Figure 10.29. A cross-section view of the meshed Rohsenow tube geometry.

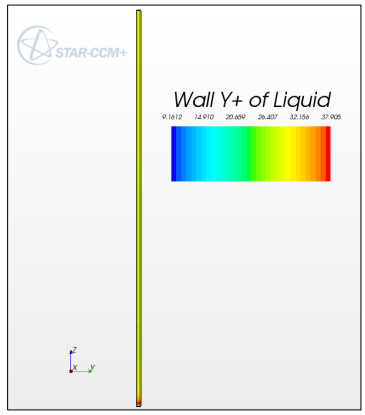


Figure 10.30. The wall y^+ values of the Rohsenow test section for an inlet velocity of 3.058 m/s.

$$N'' = -37842.5\left(1 - e^{4.22513\Delta T_{sup}}\right) \tag{10.7}$$

The heat fluxes tested for a pressure of 2000 psi and an inlet velocity of 3.048 m/s were 3.41, 4.07, 4.61, and 5.11 MW/m^2. The inlet temperature of the test section was calculated for each heat flux using the linear heat generation and the bulk temperature provided at thermocouple position 2. The calculated inlet temperature is shown in Table 10.6.

Table 10.6. Inlet temperature for the Rohsenow test cases at 2000 psi and an inlet velocity of 3.048 m/s.

Heat Flux [MW/m^2]	Inlet ΔT_{sub} [K]
3.41	130.7
4.07	130.9
4.61	131.0
5.11	136.4

In the bubble departure diameter models, the advancing and receding contact angles used for the nickel test section were 79° and 34° respectively [119]. For the calculation of the quenching component of the new model, the density of the heater was 8900 kg/m^3 and the heat capacity of the heater was 0.444 J/g-K [120].

The wall temperature experimental results were to be compared against both the standard wall boiling model and the new wall boiling model predictions. The experimental results for each heat flux was averaged over a combination of the thermocouples at positions 2, 3, 4, 5, and 6. The standard wall boiling model diverged for all heat fluxes and inlet subcoolings tested. As a result, only the wall temperature predictions for the new model (Gilman) are compared to the experimental data in Table 10.7. In the test cases with heat fluxes of 3.41, 4.07, and 5.11 MW/m^2, the new model predicts wall temperatures within the experimental error of the data.

The significance of each partitioned component of the model in the overall heat removal from the surface was also investigated for the Rohsenow test case with the highest heat flux (5.11 MW/m^2) and the results are shown in Figure 10.31. These results show that the forced convection and sliding conduction (encompassing all transient conduction on the heater) components compete for the greatest effect on heat removal near the tube inlet where the liquid is highly subcooled. The forced convection component starts as the largest component and then decreases to about zero near the outlet. While the bulk liquid remains subcooled through the tube, the sliding conduction component increases quickly while the quenching and

Table 10.7. Inlet temperature for the Rohsenow test cases at 2000 psi and an inlet velocity of 3.048 m/s.

Heat Flux [MW/m^2]	Thermocouple Positions	Experiment ΔT_{sup} [K]	Experiment Error [K]	Gilman (2014) ΔT_{sup} [K]
3.41	5,6	2.58	+/- 1.7	0.98
4.07	2,3,5,6	2.65	+/- 1.7	1.10
4.61	2-6	3.92	+/- 1.7	1.02
5.11	2-6	4.14	+/- 1.7	2.36

evaporation components increase very slowly. Once the active nucleation site density nears the maximum value, and the liquid nears saturation temperature near the outlet, the contribution towards heat removal by the evaporation component quickly increases while the sliding conduction decreases.

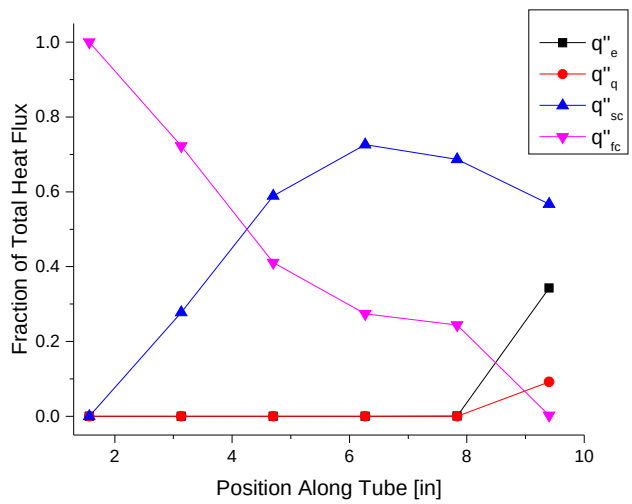


Figure 10.31. The fraction of the total heat flux removed by each component in the new model's heat partitioning model for the Rohsenow test case with heat flux of 5.11 MW/m^2 at a pressure of 2000 psi.

11 Model Sensitivity Studies

The solutions to the simulations are sensitive to parameters employed in the wall boiling models, and it is important to understand how the use of correlations may be limited to a specific set of conditions. A sensitivity study on the heat partitioning used in STAR-CCM+ was completed by Asher et al. [36] by varying parameters used in the closure models to determine their effect on the prediction of the pressure drop, wall temperature, and void fraction profile. All the parameters included in this study are shown in Table 11.1. The parameters that had the strongest effect on the results were C_L, D_d, and C_μ.

Table 11.1. Parameters used in full sensitivity study [36].

Parameter	Symbol
Lift Coefficient	C_L
Drag Coefficient	C_D
Virtual Mass Coefficient	C^{VM}
Bubble Departure Diameter	D_d
Turbulent Viscosity Scaling	C_μ
Liquid to Interface Nusselt Number	Nu_l
Gas to Interface Nusselt Number	Nu_g

In this work, sensitivity studies were completed on the parameters of the new model that have been identified to have a significant impact on the resulting boiling curve predictions. Therefore, a sensitivity study was completed on the nucleation site density model for both the standard and new wall boiling models to observe the effects on the wall temperature predictions when using the Lemmert-Chawla and Hibiki-Ishii models. The dependence on the bubble departure frequency for the wall temperature prediction was also tested for a range of frequencies. In addition, the sensitivity of the new wall boiling model to the maximum Eo number was completed. The maximum Eo number defines the largest bubble lift-off diameter allowed, given the fluid properties, and is employed if the mechanistic bubble lift-off diameter model provides a value that is greater than this defined maximum D_l.

In addition to the three terms that were identified to most likely impact the results, sensitivity studies were completed on the effect of the internal grid size, the prism layer thickness, the assumed bubble hot spot temperature change (ΔT_h) employed in the quenching component, and the assumed maximum thickness (δ_{max}) of the microlayer used in the evaporation component. These sensitivity studies were completed on the boiling curves for the MIT flow boiling facility.

11.1 Active Nucleation Site Density Models

The effect of employing different nucleation site density models on the resulting boiling curves was investigated using data from the MIT flow boiling facility for a mass flux of 500 kg/m^2-s, a pressure of 1.05 bar, and an inlet subcooling of 10 K. The boiling curve was calculated for the standard set of models, shown in Table 3.5, using (1) the Lemmert-Chawla and (2) the Hibiki-Ishii nucleation site density model while holding all other models the same. The results are compared in Figure 11.1. The wall temperatures were also calculated for the new wall boiling model using the two nucleation site density models and compared to the experimental data in Figure 11.2.

The comparison of the wall temperature predictions when using the two nucleation site density models illustrates the need to use a nucleation site density model that accurately captures the sharp increase in active nucleation sites at higher wall superheats. Since the Hibiki-Ishii model does capture this physical

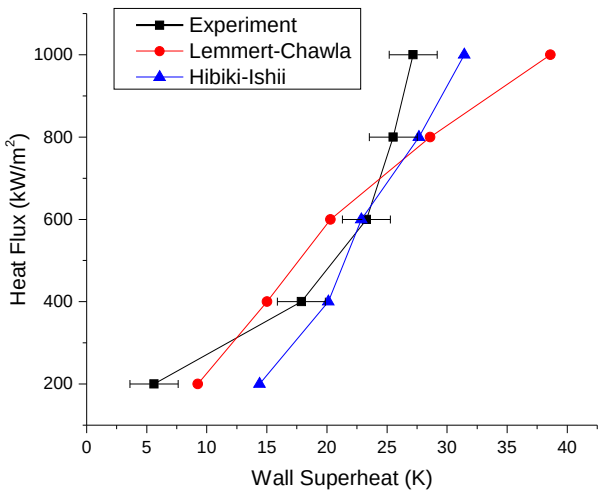


Figure 11.1. Boiling curves calculated using the standard set of models and compared to the experimental data for a mass flux of 500 kg/m^2-s, a pressure of 1.05 bar, and an inlet subcooling of 10 K.

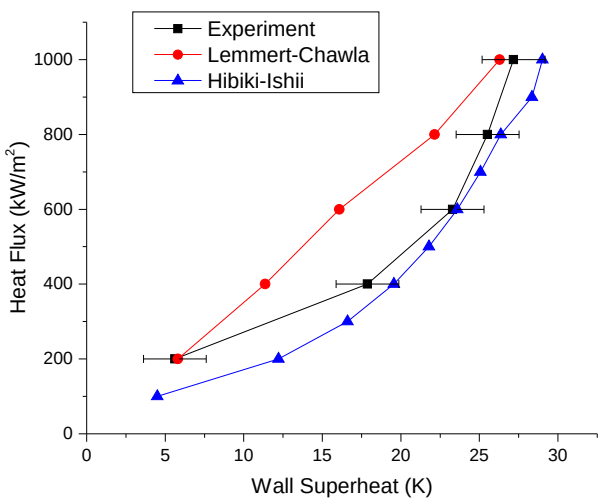


Figure 11.2. Boiling curves calculated using the new wall boiling model and compared to the experimental data for a mass flux of 500 kg/m^2-s, a pressure of 1.05 bar, and an inlet subcooling of 10 K.

phenomenon better than Lemmert-Chawla, the use of the Hibiki-Ishii model provides results that more accurately capture the shape of the boiling curves.

11.2 Bubble Departure Frequency

The Cole (1960) bubble departure frequency, shown in (3.30), is highly dependent on the bubble departure diameter, and the prediction of the bubble departure diameter remains relatively constant for a given mass flux when using the mechanistic force-balance model based on the work by Sugrue et al. Therefore, the Cole frequency prediction is approximately constant for all heat fluxes for a given mass flux and system pressure. The influence of the departure frequency on the wall temperature predictions was investigated by employing constant frequency values of 50, 150, 500, and 750 Hz on the test cases having a mass flux of 500 kg/m^2-s, inlet subcooling of 10 K, and pressures of 1.5 and 2.0 bar. For the test case at 2.0 bar, the bubble departure frequency predicted by Cole was ~250 Hz and the results are shown in Figure 11.3.

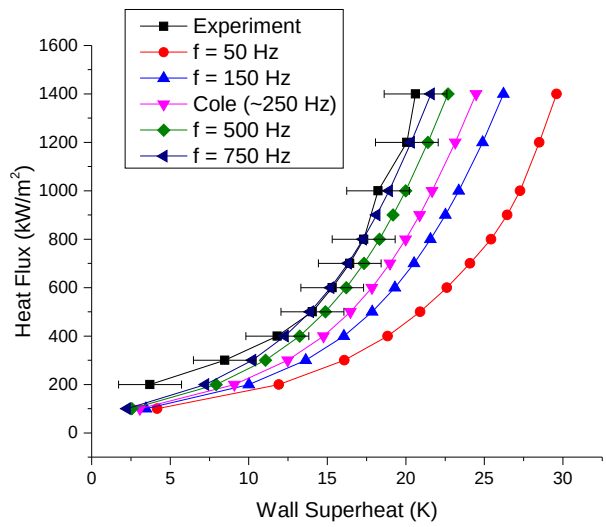


Figure 11.3. Wall temperature predictions calculated using the new wall boiling model with varying bubble departure frequencies, and compared to the experimental data for a mass flux of 500 kg/m^2-s, a pressure of 2.0 bar, and an inlet subcooling of 10 K.

The results illustrate that at very low bubble departure frequencies, a change in the frequency value has a much larger impact on the temperature predictions. As the value of the frequency increases, the relative temperature change between the curves becomes smaller when the frequency value changes. For this test case, the average experimental frequency was ~400-450 Hz, although the frequency calculated from experiments does vary with the heat flux. The results in Figure 11.3 show that the boiling curve with a constant bubble departure frequency of 500 Hz, which is closer to the recorded experimental frequency, predicts the wall temperature closer to the experimental data than the boiling curve using the Cole bubble departure frequency model.

The results of the test case at a pressure of 1.5 bar are shown in Figure 11.4, where the bubble departure frequency predicted by Cole was ~240 Hz. In this test case, the experimental bubble departure frequency was measured at ~75-150 Hz (the bubble departure frequency increased with increasing heat flux). The boiling curves produced using frequencies of 50 and 150 Hz are closer to the experimental boiling curve than the boiling curve produced by using the Cole bubble departure frequency model.

The sensitivity of the wall temperature predictions on the value of the bubble departure frequency indicate that an improved and more accurate frequency model would likely increase the accuracy of the new wall

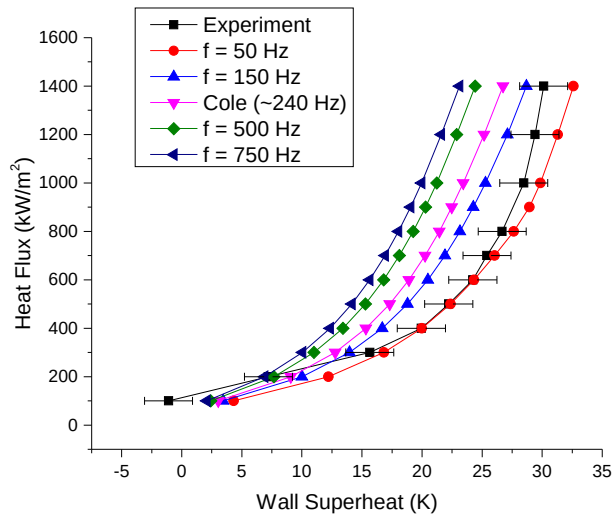


Figure 11.4. Wall temperature predictions calculated using the new wall boiling model with varying bubble departure frequencies, and compared to the experimental data for a mass flux of 500 kg/m^2-s, a pressure of 1.5 bar, and an inlet subcooling of 10 K.

boiling model's predictions. The new wall boiling model is sensitive to the bubble departure frequency because it is directly included in the calculation of the evaporation, sliding conduction, and quenching terms.

11.3 Maximum Eötvös number

The maximum Eo number employed in the new wall boiling model predicts the maximum size that a bubble can reach before deviating from a spherical form. It is assumed that once the bubble deviates from its spherical shape, it is driven away from the wall by the increased lift-force caused by its non-spherical shape. The larger the maximum Eo number, the larger the maximum bubble lift-off diameter can be. As the bubble lift-off diameter increases, the sliding conduction heat transfer coefficient also increases since the bubbles slide a longer distance before reaching the lift-off diameter size. This increased heat transfer coefficient is seen the results by predicting lower wall temperatures for a set of conditions.

The maximum Eo is an important component to the new model because the mechanistic bubble departure model for the lift-off diameter can predict unphysically large diameters at high wall superheats. To observe the effect of different maximum Eo numbers on the wall temperature predictions, the maximum Eo value imposed was varied from 0.05 to 2.0 (the nominal value for the model is 0.1) and compared to the results from the MIT Flow Boiling Facility. The results for a mass flux of 500 kg/m^2-s, 15 K subcooling, and a pressure of 2.0 bar are shown in Figure 11.5. The results for varied maximum Eo numbers for a mass flux of 1000 kg/m^2-s, 15 K subcooling, and a pressure of 2.0 bar are shown in Figure 11.6. The results at a subcooling of 15 K illustrate that a lower maximum Eo number (0.05 to 0.1) tends to predict the wall temperatures more closely to the experimental data. At this higher subcooling, the thermal boundary layer where the bubbles grow tends to have a sharper decrease in temperature. Therefore, smaller lift-off diameters are expected.

The effect of the maximum Eo number on a subcooling of 10 K (at 2.0 bar) is shown in Figure 11.7 for a mass flux of 500 kg/m^2-s and Figure 11.8 for a mass flux of 1000 kg/m^2-s. These figures illustrate that the intermediate value for the maximum Eo number (0.5 and 1.0) predict the wall temperatures closer to the experimental data.

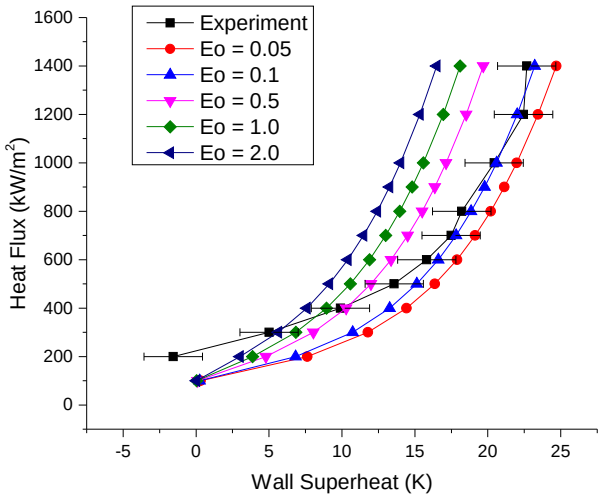


Figure 11.5. Average heater temperature predictions for the experiment and the new model with varied maximum Eo numbers for a mass flux of 500 kg/m^2-s, inlet subcooling of 15 K, and pressure of 2.0 bar.

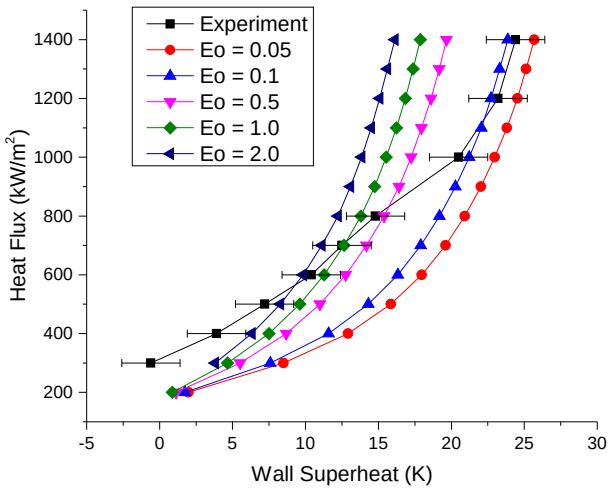


Figure 11.6. Average heater temperature predictions for the experiment and the new model with varied maximum Eo numbers for a mass flux of 1000 kg/m^2-s, inlet subcooling of 15 K, and pressure of 2.0 bar.

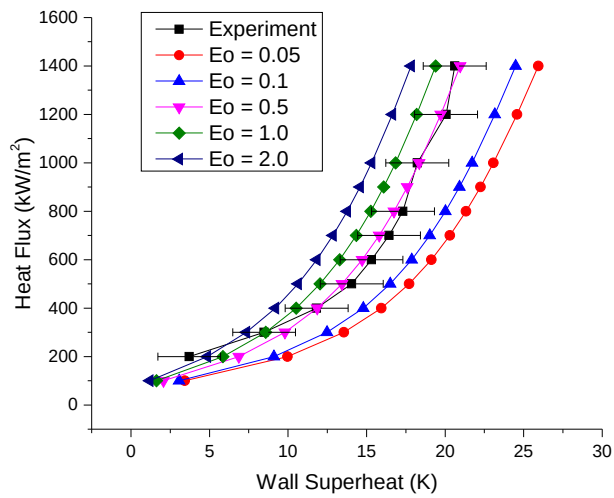


Figure 11.7. Average heater temperature predictions for the experiment and the new model with varied maximum Eo numbers for a mass flux of 500 kg/m^2-s, inlet subcooling of 10 K, and pressure of 2.0 bar.

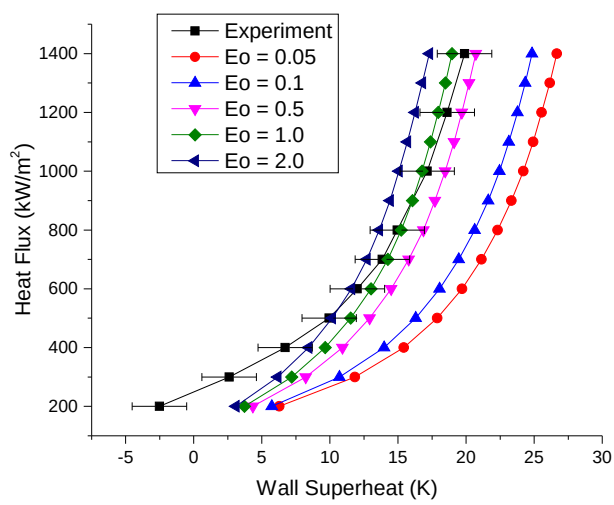


Figure 11.8. Average heater temperature predictions for the experiment and the new model with varied maximum Eo numbers for a mass flux of 1000 kg/m^2-s, inlet subcooling of 10 K, and pressure of 2.0 bar.

When the degree of subcooling at the inlet is decreased to 5 K at a pressure of 2.0 bar, even larger values of the maximum Eo number (1.0 to 2.0) predict wall temperatures more closely to the experimental data. This is shown in Figure 11.9 for a mass flux of 500 kg/m^2-s and Figure 11.10 for a mass flux of 1000 kg/m^2-s. The maximum Eo number plots clearly illustrate the effect that the maximum allowed bubble lift-off diameter has on the prediction of the wall superheat.

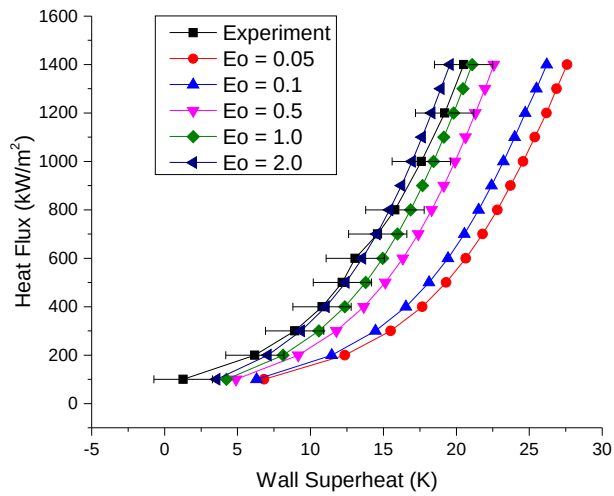


Figure 11.9. Average heater temperature predictions for the experiment and the new model with varied maximum Eo numbers for a mass flux of 500 kg/m^2-s, inlet subcooling of 5 K, and pressure of 2.0 bar.

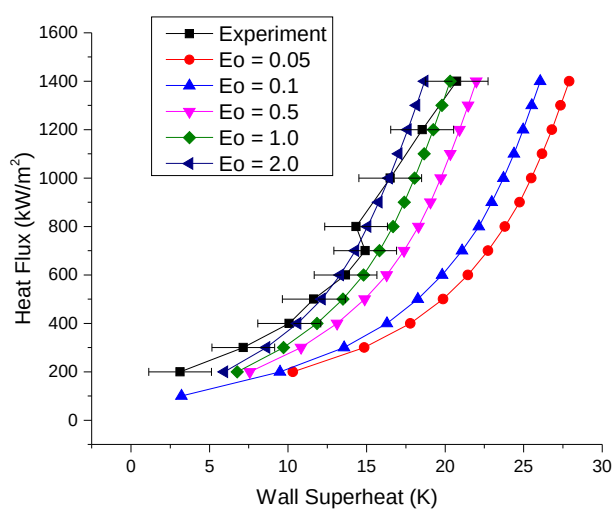


Figure 11.10. Average heater temperature predictions for the experiment and the new model with varied maximum Eo numbers for a mass flux of 1000 kg/m^2-s, inlet subcooling of 5 K, and pressure of 2.0 bar.

The effect of the maximum Eo number was also studied on the test cases with mass fluxes of 500 and 1000 kg/m^2-s with 10 K subcooling and at a pressure of 1.5 bar to compare to the sensitivity study completed at 2.0 bar. The comparison of the boiling curves for different maximum Eo numbers are shown in Figures 11.11 and 11.12 for a pressure of 1.5 bar. In contrast to the results at 2.0 bar (shown in Figures 11.7 and 11.8),

these figures show that at a pressure of 1.5 bar, lower maximum Eo numbers produce a boiling curve that is closer to the experimental data.

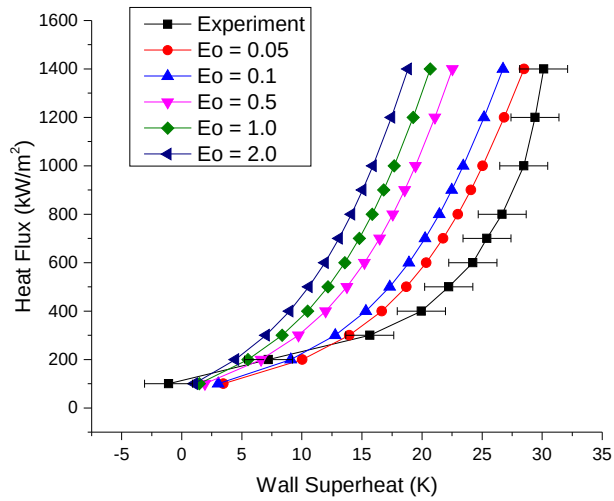


Figure 11.11. Average heater temperature predictions for the experiment and the new model with varied maximum Eo numbers for a mass flux of 500 kg/m^2-s, inlet subcooling of 10 K, and pressure of 1.5 bar.

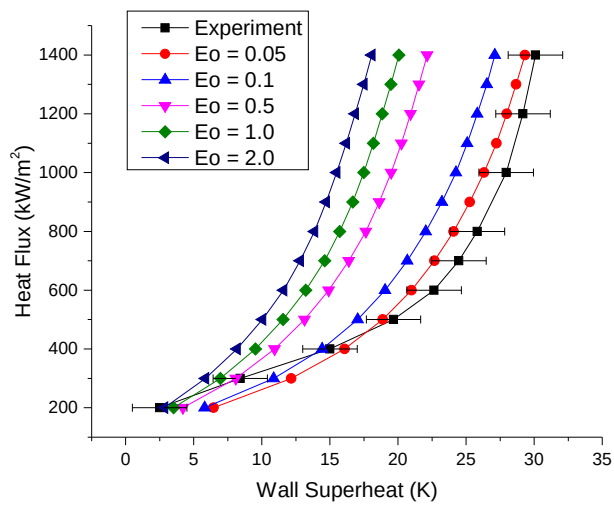


Figure 11.12. Average heater temperature predictions for the experiment and the new model with varied maximum Eo numbers for a mass flux of 1000 kg/m^2-s, inlet subcooling of 10 K, and pressure of 1.5 bar.

11.4 Grid Size

The new subcooled boiling model's wall temperature predictions dependence on the cell size was investigated at a mass flux of 500 kg/m^2-s, inlet subcooling of 10 K, pressure of 2.0 bar, and heat flux of 1200 kW/m^2. In the first study, the near wall prism layer was held at a constant thickness of 0.9 mm so that the wall y^+ value remained at $\sim$43. The base cell size of the remaining internal mesh was then varied by

employing a constant multiplication factor of 2. The results are shown in Table 11.2 and indicate that the new wall boiling model has little dependence on the internal cell size of the mesh.

Table 11.2. Liquid heat transfer coefficient and wall superheat predictions with varied base cell size.

Base Cell Size [mm]	Prism Layer Thickness [mm]	h_l [kW/m^2-K]	ΔT_{sup} [K]
1.8	0.9	36.7	23.19
0.9	0.9	36.6	23.17
0.45	0.9	36.3	23.16

An additional grid study was completed on the dependence on the thickness of the prism layer with the base cell size held at a constant 0.9 mm. A constant multiplication factor of 2 on the size was employed, but at the smallest prism layer thickness, the value was increased slightly to retain mesh integrity. The wall y$^+$ values were recorded for each size, and were all very close to being in the desired range of 30-100. The results are shown in Table 11.3 for the new model's calculation of the liquid heat transfer coefficient. These results indicate that the thickness of the near wall prism layer does not effect the liquid heat transfer coefficient.

Instead, the wall temperature predictions are sensitive to the thickness of the near wall prism layer. This dependence on the prism layer is to be expected due to the method in which the software STAR-CCM+ calculates the heat flux removed by each component of the model. As described in Section 8, the removal of heat through each component is completed by using coefficients that are dependent on both the wall temperature and the temperature of the wall cell. If the near wall cell is smaller in size (smaller prism layer thickness), the average temperature of the cell is a higher value because of the thermal boundary layer near the wall. Since the heat transfer coefficient calculations do not change with changes in cell size, and the average cell temperature near the wall increases with a smaller cell size, this results in a higher wall temperature. As a result, the thickness of the near wall prism layer should be maintained to have a wall y$^+$ near 30.

Table 11.3. Liquid heat transfer coefficient and wall superheat calculated with varied prism layer thickness.

Base Cell Size [mm]	Prism Layer Thickness [mm]	Wall y$^+$	h_l [kW/m^2-K]	ΔT_{sup} [K]
0.9	0.5	28	37.3	25.76
0.9	0.9	43	36.6	23.17
0.9	1.8	108	36.3	22.35

11.5 Bubble Dry Spot Temperature Change

The calculation of the quenching component in the new heat partitioning model, described in Section 6.3, is dependent on the assumption of the temperature difference between the bubble hot spot (the dry surface area under the bubble) and the surrounding average heater temperature not under the influence of the bubble (ΔT_h). Experiments have shown that this hot spot is typically on the order of a few degrees. A sensitivity study was completed on the quenching component by varying ΔT_h, using values of 1, 2 (nominal value employed in the new model), 4, 8, and 20 K on the test cases with a mass flux of 500 kg/m^2-s, inlet subcooling of 10 K, and pressures of 1.5 and 2.0 bar. The results on the wall temperature predictions at 2.0 bar are shown in Figure 11.13 and indicate that even when ΔT_h is varied by an order of magnitude, it has very little effect on the wall temperature prediction. A similar result was seen for a pressure of 1.5

bar, which is shown in Figure 11.14. The very low sensitivity to the value of ΔT_h was expected since the quenching component is the smallest contributor to heat removal from the heater surface.

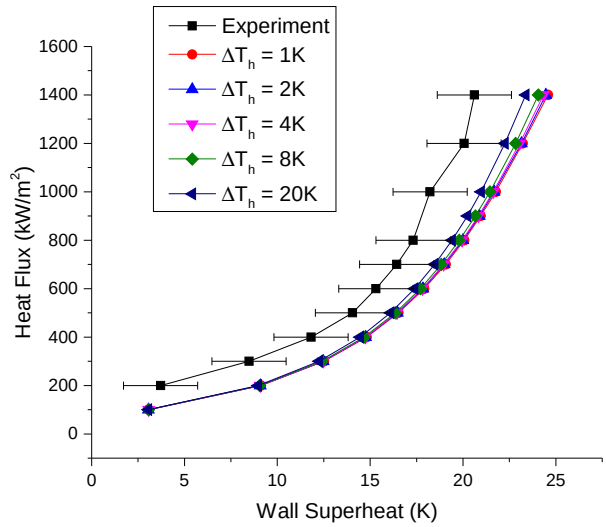


Figure 11.13. Wall temperature predictions calculated using the new wall boiling model with varying bubble hot spot temperatures, and compared to the experimental data for a mass flux of 500 kg/m^2-s, a pressure of 2.0 bar, and an inlet subcooling of 10 K.

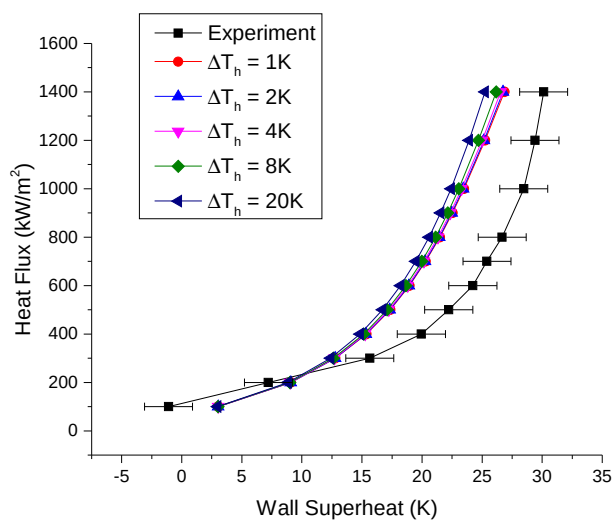


Figure 11.14. Wall temperature predictions calculated using the new wall boiling model with varying bubble hot spot temperatures, and compared to the experimental data for a mass flux of 500 kg/m^2-s, a pressure of 1.5 bar, and an inlet subcooling of 10 K.

11.6 Maximum Microlayer Thickness

In the evaporation component of the new heat partitioning model, the calculation of the microlayer evaporation is based on the assumption that the microlayer has a given maximum thickness (δ_{max}). The sensi-

tivity of the wall temperature predictions to the value of δ_{max} was studied by implementing the values of 0.5, 1, 2 (nominal value employed in the new model), and 4 µm on the test cases that have a mass flux of 500 kg/m^2-s, inlet subcooling of 10K, and pressures of 1.5 and 2.0 bar. The smallest microlayer thickness is limited to being on the order of hundreds of nm, because when the liquid layer trapped by the bubble is nanometers in thickness the attractive inter-molecular forces between the solid and liquid prevent evaporation. This means that the liquid layer would actually be an adsorbed film and not a microlayer. The maximum thickness is limited to being no larger than a few microns so that the liquid layer remains a true microlayer and is not a part of the macrolayer of the bubble [102, 121].

The results of the sensitivity study are shown in Figures 11.15 and 11.16 for pressures of 2.0 and 1.5 bar respectively. These figures show that the wall temperature predictions essentially do not change when the microlayer thickness is varied from 0.5-4 µm.

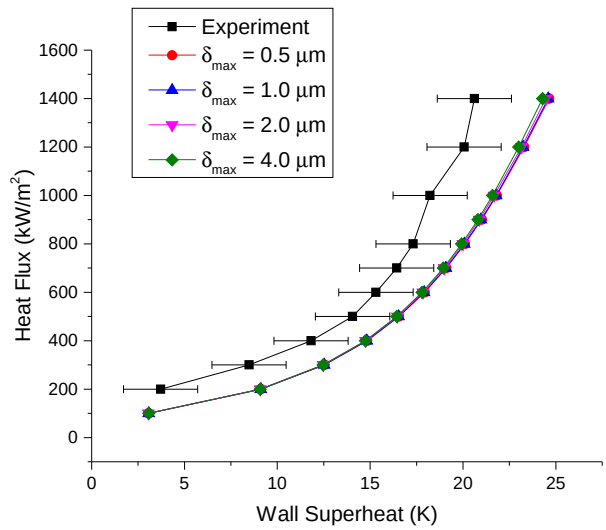


Figure 11.15. Wall temperature predictions calculated using the new wall boiling model with varying maximum microlayer thickness, and compared to the experimental data for a mass flux of 500 kg/m^2-s, a pressure of 2.0 bar, and an inlet subcooling of 10 K.

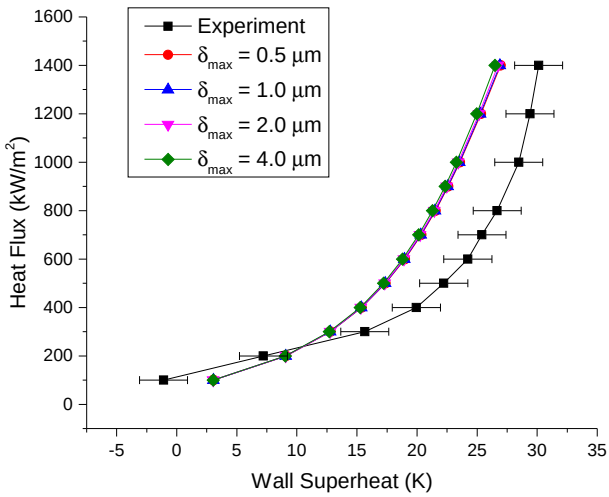


Figure 11.16. Wall temperature predictions calculated using the new wall boiling model with varying maximum microlayer thickness, and compared to the experimental data for a mass flux of 500 kg/m^2-s, a pressure of 1.5 bar, and an inlet subcooling of 10 K.

12 Summary, Conclusions, and Future Work

12.1 Summary

CFD has become a method of numerical analysis that can aid experiments for studies to optimize flow conditions and new or improved heat transfer designs. In particular, the CASL program aims to have a virtual environment for reactor analysis, which provided the goal of this work to develop a wall boiling model that can predict the wall temperatures of the heated rods in a nuclear reactor while correctly describing the phenomena on the heated surface to provide insight on CHF for a given system. A better understanding of when CHF will occur could allow for greater power generation while ensuring a sufficient operating safety margin.

In many industrial applications, including LWRs, subcooled flow boiling is used when a high heat transfer coefficient is desired. In this work, a more general boiling closure model for subcooled flow boiling conditions was developed and tested at low and high pressures. The model includes all physical boiling phenomena and necessary mechanisms to accurately predict the temperature and heat flux at the wall in CFD simulations and was implemented using STAR-CCM+. Experiments have illustrated that understanding and predicting the interactions on the heated surface during boiling can aid in the prediction of the wall temperature and CHF. The model explicitly tracks the dry surface area during boiling so that the model can be further extended to predict DNB by studying the fraction of dry surface area as the heat flux increases for a system.

12.2 Conclusions

The conclusions based on the numerical testing of the new wall boiling model in STAR-CCM+ are summarized below.

- The wall temperature predictions using the new model (Gilman 2014) are slightly closer to experimental data (and typically very near or within experimental error) than the standard (Kurul-Podowski) model for mass fluxes of 500, 1000, and 1250 kg/m^2-s, a subcooling of 10 K, and at a pressure of 1.05 bar. The new model also captures the shape of the boiling curves more accurately, while being able to converge at all heat fluxes tested by the experiment.
- The wall temperature predictions using the new model also capture the shape of the boiling curves better than the standard model for mass fluxes of 500, 1000, and 1250 kg/m^2-s, a subcooling of 10 K, and at a pressure of 1.5 bar. The standard model predicts wall temperatures that are closer to experimental data at this pressure, while the new model tends to slightly under-predict the wall temperatures. The standard model does not converge at higher heat fluxes.
- Both the new and standard models over-predict the wall temperatures for mass fluxes of 500, 1000, and 1250 kg/m^2-s, a subcooling of 5 K, and a pressure of 2.0 bar. The new model is closer to experimental data and is more robust at higher heat fluxes.
- Both the new and standard models over-predict the wall temperatures for mass fluxes of 500, 1000, and 1250 kg/m^2-s, a subcooling of 10 K, and a pressure of 2.0 bar. The new model is closer to the experimental data than the standard model and is very near to being within the experimental error, while converging for all heat fluxes tested.
- The new model is very close to the experimental wall temperatures for mass fluxes of 500, 1000, and 1250 kg/m^2-s, a subcooling of 15 K, and pressure of 2.0 bar. The new model is within the experimental error for almost all points in the boiling curves for mass fluxes of 500 and 1250 kg/m^2-s and is within experimental error for the high heat fluxes for the case with a mass flux of 1000 kg/m^2-s. The standard model over-predicts the wall temperatures in all cases.

- The new model predicts wall temperatures within the experimental error for almost all the cases tested at high pressure (2000 psi). The standard model was unable to converge at the high heat fluxes and subcoolings of these test cases.
- The new model captures the increase in dry surface area as the heat flux increases, while including a dependence on the heater properties through the use of the Hibiki-Ishii nucleation site density model, the mechanistic bubble departure diameter models, and the quenching component.
- The heat partitioning components of the new model show that sliding (transient) conduction and forced convection are the highest modes of heat removal from the surface at low pressure, relatively low active nucleation site densities ($\lesssim 10^7$ sites/m^2), and with subcooled liquid in the channel. The forced convection component decreases to zero as the other components increase with increased boiling. Both the quenching and evaporation components steadily rise with increased heat flux and have lower contributions to the heat removal.
- At high pressure, the heat partitioning components of the new model show that sliding (transient) conduction and forced convection dominate as the methods of heat removal while the liquid remains highly subcooled. Also, the forced convection component decreases as the amount of wall boiling increases and sliding conduction becomes the dominant mode of heat transfer. Once the active nucleation site density increases to relatively high values ($\gtrsim 10^{11}$ sites/m^2) and the temperature of the liquid nears the saturation temperature (near the tube outlet), the evaporation component quickly increases and the sliding conduction component decreases but remains the dominant mode of heat removal.
- The new model's prediction of the wall superheat is dependent on the value of the bubble departure frequency, particularly when the frequency value is $\lesssim 500$ Hz. When a bubble departure frequency value near the experimental value is implemented, the resulting boiling curves are closer to the experimental data.
- The new model is sensitive to the maximum Eo number because it defines the maximum bubble lift-off diameter. This effects the amount of transient conduction that occurs on the surface from the disrupted thermal boundary layer because a larger bubble diameter corresponds to a larger area of influence for the transient conduction.
- The new model's predictions of both the wall superheat and the liquid heat transfer coefficient are insensitive to the internal cell size of the mesh when the prism layer thickness is held constant.
- The new model's prediction of the liquid heat transfer coefficient also has very little sensitivity to the prism layer thickness. The wall superheat prediction has some sensitivity to the prism layer thickness because of the method the heat transfer coefficient is passed into the STAR-CCM+ software.
- The prediction of the wall superheat by the new model has very little dependence on the value of ΔT_h (used in the quenching component).
- The prediction of the wall superheat by the new model does not have a dependence on the value of the maximum microlayer thickness (δ_{max}) when reasonable physical values are implemented.
- Both the new and standard models have a dependence on the nucleation site density model implemented. The wall boiling models should have a dependency on the nucleation site density because it helps capture the effects of the material properties of the heater.
- Overall, the new model predicts wall temperatures closer to experimental data at both low and high pressures and also converges at higher heat fluxes and greater subcoolings than the standard model. All model parameters are theoretical based, so even though the model has sensitivity to some parameters, it has not been adjusted to fit the results.

12.3 Suggested Future Work

- Continued testing as additional wall temperature data for water at high pressure becomes available.
- Continued evaluation of closure models as new data and information become available. In particular, the bubble departure model could be improved to include a dependence on the wall superheat on the bubble size.
- Re-evaluation of the microlayer evaporation assumptions as studies on the microlayer are completed for flow boiling.
- Implementation of a bubble departure frequency model that incorporates a heater material dependent wait time, since t_w is physically dependent on the thermal response of the heater.
- The model formulation can be re-investigated to evaluate if it can be tested and provide accurate predictions of wall temperatures on non-ideal surfaces, such as surfaces with corrosion product deposits.
- Use of the dry surface area calculation and bubble merging probability in DNB testing.
- After all physical phenomena at the wall are verified, the void fraction predictions across the channel can be investigated.

References

[1] U.S. Department of Energy, "Consortium for advanced simulation of LWRs," 2010.

[2] N. E. Todreas and M. S. Kazimi, *Nuclear Systems: Thermal Hydraulic Fundamentals; Vol 1, 2nd Ed.* CRC Press, 2012.

[3] V. P. Carey, *Liquid-Vapor Phase-Change Phenomena, 2nd Ed.* Taylor and Francis Group, 2008.

[4] A. Sanna, C. Hutter, D. Kenning, T. Karayiannis, K. Sefiane, and R. Nelson, "Nucleate pool boiling investigation on a silicon test section with micro-fabricated cavities," in *ECI International Conference on Boiling Heat Transfer*, (Florianopolis, Brazil), May 3-7 2009.

[5] E. Hahne and U. Grigull, eds., *Heat Transfer in Boiling*. Academic Press and Hemisphere, 1977.

[6] J. G. Collier and J. R. Thome, *Convective boiling and condensation, 3rd Ed.* Oxford University Press, 1994.

[7] D. D. Hall and I. Mudawar, "Critical heat flux (CHF) for water flow in tubes- i. compliation and assessment of world CHF data," *International Journal of Heat and Mass Transfer*, vol. 43, pp. 2573–2604, 2000.

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

[9] A. Guion, "Liquid microlayer on a heating wall," Master's thesis, Institut National Des Sciences et Techniques Nucleaires, 2011.

[10] V. Mizo, "Investigation of intertia controlled bubble departure mechanism in subcooled flow boiling using high speed photography," Master's thesis, Rochester Institute of Technology, 1995.

[11] H. Kim and J. Buongiorno, "A novel infrared-based experimental technique to detect phase dynamics on boiling surfaces," *NURTH-14*, pp. 1–10, 2011.

[12] Y. B. J. Kim, Hyungdae; Park, "Measurement of wetted area fraction in subcooled pool boiling of water using infrared thermography," *Nuclear Engineering and Design*, vol. 264, pp. 103–110, 2013.

[13] T. H. Kim, E. Kommer, S. Dessiatoun, and J. Kim, "Measurement of two-phase flow and heat transfer parameters using infrared thermometry," *International Journal of Multiphase Flow*, vol. 40, no. 0, pp. 56 – 67, 2012.

[14] C. Gerardi, *Investigation of the pool boiling heat transfer enhancement of nano-engineered ffluid by mmean of high-speed infrared thermography*. PhD thesis, Massachusetts Institute of Technology, 2009.

[15] 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 Transfer*, vol. 48, no. 25–26, pp. 5536 – 5548, 2005.

[16] 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. 53, no. 19–20, pp. 4185 – 4192, 2010.

[17] 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.

[18] Y.-H. Zhao, T. Masuoka, and T. Tsuruta, "Unified theoretical prediction of fully developed nucleate boiling and critical heat flux based on a dynamic microlayer model," *International Journal of Heat and Mass Transfer*, vol. 45, no. 15, pp. 3189 – 3197, 2002.

[19] R. A. Knief, *Nuclear Engineering-:Theory and Technology of Commercial Nuclear Power, 2nd. Ed.* American Nuclear Society, Inc., 2008.

[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, pp. 2245–2259, 1994.

[22] R. Sugrue, "The effects of orientation angle, subcooling, heat flux, mass flux, and pressure on bubble growth and detachment in subcooled flow boiling," Master's thesis, Massachusetts Institute of Technology, 2012.

[23] 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.

[24] 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, no. 23, pp. 3857 – 3871, 1998.

[25] S. Lin, 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.

[26] 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, no. 11–12, pp. 2769 – 2778, 2012.

[27] 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.

[28] V. Prodanovic, D. Fraser, and M. Salcudean, "Bubble behavior in subcooled flow boiling of water at low pressures and low flow rates," *International Journal of Multiphase Flow*, vol. 28, no. 1, pp. 1 – 19, 2002.

[29] G. Sateesh, S. K. Das, and A. Balakrishnan, "Analysis of pool boiling heat transfer: effect of bubbles sliding on the heating surface," *International Journal of Heat and Mass Transfer*, vol. 48, no. 8, pp. 1543 – 1553, 2005.

[30] 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," *NURETH-15*, Pisa, Italy, May 12-17, 2013.

[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, no. 17–18, pp. 3659 – 3667, 2004.

[32] V. H. DelValle and D. Kenning, "Subcooled flow boiling at high heat flux," *International Journal of Heat and Mass Transfer*, vol. 28, no. 10, pp. 1907 – 1920, 1985.

[33] T. Theofanous and T. Dinh, "High heat flux boiling and burnout as microphysical phenomena: mounting evidence and opportunities," *Multiphase Science and Technology*, vol. 18 (3), pp. 251–276, 2006.

[34] C. Gerardi, J. Buongiorno, L. Hu, and T. McKrell, "Infrared thermometry study of nanofluid pool boiling phenomena," *Nanoscale Research Letters*, vol. 6:232, 2011.

[35] 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.

[36] I. M. Asher, T. J. Drzewiecki, T. P. Grunloh, V. E. Petrov, K. J. Fidkowski, and A. Manera, "Sensitivity, verification, and validation studies of star-cd, star-ccm+, and nphase cfd boiling models," tech. rep., CASL Milestone L3-THM-CFD-P5-03, September 29, 2012.

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

[38] G. Kocamustafaogullari, "Pressure dependence of bubble departure diameter for water," *International Communications in Heat and Mass Transfer*, vol. 10, no. 6, pp. 501 – 509, 1983.

[39] R. Cole, "A photographic study of pool boiling in the region of the critical heat flux," *American Institute of Chemical Engineering Journal*, vol. 6, pp. 533–542, 1960.

[40] 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.

[41] N. Kolev, "How accurately can we predict nucleate boiling?," *Experimental Thermal and Fluid Science*, vol. 10, no. 3, pp. 370 – 378, 1995.

[42] T. Hibiki and M. Ishii, "Active nucleation site density in boiling systems," *International Journal of Heat and Mass Transfer*, vol. 46, no. 14, pp. 2587 – 2601, 2003.

[43] 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.

[44] B. Phillips, J. Buongiorno, and T. McKrell, "Nucleation site density, bubble departure diameter, bubble departure frequency, and local temperature distribution in subcooled flow boiling of water," tech. rep., Massachusetts Institute of Technology, 2013.

[45] B. Phillips, T. McKrell, and J. Buongiorno, "Preliminary data from mit subcooled flow boiling facility," tech. rep., Massachusetts Institute of Technology, 2012.

[46] E. Baglietto and H. Ninokata, "Anisotropic eddy viscosity modeling for application to industrial engineering internal flows," *International Journal of Transport Phenomena*, vol. 8, 2006.

[47] J. Anderson, Jr, *Computational Fluid Dynamics, 3rd ed.* Springer, 2009.

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

[49] B. Launder and D. Spalding, "The numerical computation of turbulent flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 3, no. 2, pp. 269 – 289, 1974.

[50] J. H. Ferziger and M. Peric, *Computational methods for fluid dynamics, 3rd Ed.* Springer, 2002.

[51] K. Shirvan, E. Baglietto, and M. Kazimi, "Assessment of a baseline two phase CFD closure for PWR applications," in *Transactions of the American Nuclear Society*, Washington D.C, 2013.

[52] R. L. Judd and K. S. Hwang, "A comprehensive model for nucleate pool boiling heat transfer including microlayer evaporation," *Journal of Heat Transfer*, vol. 98, no. 4, pp. 623–629, 1976.

[53] N. Kurul and M. Podowski, "Multidimensional effects in forced convection subcooled boiling," *Heat Transfer 1990: Proceedings of the Ninth International Heat Transfer Conference*, vol. 2, pp. 21–26, 1990.

[54] C. G. Bartolomei and V. M. Chanturiya, "Experimental study of true void fraction when boiling subcooled water in vertical tubes," *Thermal Engineering*, vol. 14, pp. 123–128, 1967.

[55] CD-adapco, *STAR-CCM+ Software Manual*. CD-adapco, 2013.

[56] W. E. Ranz and J. W. R. Marshall, "Evalporation from drops part i," *Chemical Engineering Progress*, vol. 48, pp. 141–146, 1952.

[57] Y. Chen and F. Mayinger, "Measurement of heat transfer at the phase interface of condensing bubbles," *International Journal of Multiphase Flow*, vol. 18, no. 6, pp. 877 – 890, 1992.

[58] G. R. Warrier, N. Basu, and V. K. Dhir, "Interfacial heat transfer during subcooled flow boiling," *International Journal of Heat and Mass Transfer*, vol. 45, no. 19, pp. 3947 – 3959, 2002.

[59] I. Park, S.-M. Kim, and I. Mudawar, "Experimental measurement and modeling of downflow condensation in a circular tube," *International Journal of Heat and Mass Transfer*, vol. 57, no. 2, pp. 567 – 581, 2013.

[60] G. H. Yeoh and J. Y. Tu, "Two-fluid and population balance models for subcooled boiling flow," *Applied Mathematical Modelling*, vol. 30, pp. 1370–1391, 2006.

[61] L. Schiller and A. Naumann, "Uber die grundlegenden berecgnnungen bei der schwerkraftaufbereitung," *VDI Zeits*, vol. 77, pp. 318–320, 1933.

[62] A. Tomiyama, I. Kataoka, I. Zun, and T. Sakaguchi, "Drag coefficients of single bbubble under normal and micro gravity conditions," *JSME International Journal, Series B*, vol. 42, pp. 472–479, 1998.

[63] A. Tomiyama, H. Tamai, I. Zun, and S. Hosokawa, "Transverse migration of single bubbles in simple shear flows," *Chemical Engineering Science*, vol. 57, no. 11, pp. 1849 – 1858, 2002.

[64] S. Lo and J. Osman, "CFD modeling of boiling flow in a PBST 5x5 bundle," *Hindawi Publishing Corporation*, pp. 8, Article ID 795935, 2012.

[65] R. T. Lahey Jr and D. A. Drew, "The analysis of two-phase flow and heat transfer using a multidimensional, four field, two-fluid model," *Nuclear Engineering and Design*, vol. 204, no. 1–3, pp. 29 – 44, 2001.

[66] H. Lamb, *Hydrodynamics*. Dover Publications, New York, NY, 1945.

[67] T. R. Auton, J. C. R. Hunt, and M. Prud'Homme, "The force exerted on a body in inviscid unstead non-uniform rotational flow," *Journal of fluid Mechanics*, vol. 197, pp. 241–257, 1988.

[68] A. Troshko and Y. Hassan, "A two-equation turbulence model of turbulent bubbly flows," *International Journal of Multiphase Flow*, vol. 27, no. 11, pp. 1965 – 2000, 2001.

[69] M. Kim, J. Tu, G. H.Yeoh., and G. C. Park, "On population balance approach for subcooled boiling flow prediction," *International Journal of Multiphase Flow*, vol. 127, pp. 253–265, 2005.

[70] 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 – 1368, 2002.

[71] H. A. Van den Vorst and P. Sonneveld, "CGSTAB, a more smoothly converging variant of CGS," tech. rep., Delft University of Technology, 1990.

[72] W. Fritz, "Berechnung des maximalvolumens von dampfblasen," *Physikalische Zeitschrift*, vol. 36, pp. 379–384, 1935.

[73] W. Fritz and Ende, "Verdampfungsvorgang nach kinematographischen aufnahmen an dampfblasen," *Physikalische Zeitschrift*, vol. 37, pp. 391–401, 1936.

[74] C. H. Wang and V. K. Dhir, "Effect of surface wettability on active nucleation site density during pool boiling of water on a vertical surface," *ASME Journal of Heat Transfer*, vol. 115, pp. 659–669, 1993.

[75] C. H. Wang and V. K. Dhir, "On the gas entrapment and nucleation site density during pool boiling of saturated water," *ASME Journal of Heat Transfer*, vol. 115, pp. 670–679, 1993.

[76] S. Witharana, B. Phillips, S. Strobel, H. D. Kim, and T. McKrell, "Bubble nucleation on nano- to micro-size cavities and posts: An experimental validation of classical theory," *Journal of Applied Physics*, vol. 112, 2012.

[77] N. Cressie and L. B. Collins, "Analysis of spatial point patterns using bundles of product density lisa functions," *Journal of Agricultural, Biological, and Environmental Statistics*, vol. 6, pp. 118–135, 2001.

[78] T. E. Smith, *Notebook for Spatial Data Analysis. Part I. Spatial Point Pattern Analysis*, ch. 3, pp. 1–27.

[79] N. Zuber, "The dynamics of vapor bubbles in nonuniform temperature fields," *International Journal of Heat and Mass Transfer*, vol. 2, no. 1–2, pp. 83 – 98, 1961.

[80] B. Yun, A. Splawski, S. Lo, and C. Song, "Prediction of a subcooled boiling flow with mechanistic wall boiling and bubble size models," tech. rep., CD-Adapco, 2010.

[81] S. M. Salim and S. Cheah, "Wall y+ strategy for dealing with wall-bounded turbulent flows," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. II, March 18-20 2009.

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

[83] H. Schlichting and K. Gersten, *Boundary Layer Theory*. Springer, 2000.

[84] 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 obstacles," tech. rep., Massachusetts Institute of Technology, 2012.

[85] ASCOMP, Technoparkstrasse 1, CH-8005 Zurich, Switzerland, *Multi-Fluid Navier-Stokes Solver TransAT user Manual*, May 2009.

[86] K. Iwamoto, Y. Suzuki, and N. Kasagi, "Dns datadata of turbulence and heat transfer," 2002.

[87] J. 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.

[88] L. Zeng, J. Klausner, and R. Mei, "A unified model for the prediction of bubble detachment diameters in boiling systems- i. pool boiling," *International Journal of Heat and Mass Transfer*, vol. 36, no. 9, pp. 2261 – 2270, 1993.

[89] 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.

[90] J. H. Lienhard IV and J. H. Lienhard V, *A Heat Transfer Textbook, 3rd Ed.* Phlogiston Press, 2008.

[91] S. Maity, "Effect of velocity and gravity on bubble dynamics," Master's thesis, University of California, Los Angeles, 2000.

[92] N. Basu, G. R. Warrier, and V. K. Dhir, "Wall heat flux partitioning during subcooled flow boiling: Part ii—model validation," *Journal of Heat Transfer*, vol. 127, no. 2, pp. 141–148, 2005.

[93] G. R. Warrier and V. K. Dhir, "Heat transfer and wall heat flux partitioning during subcooled flow nucleate boiling—a review," *Journal of Heat Transfer*, vol. 128, no. 12, pp. 1243–1256, 2006.

[94] J. Lu, S. Biswas, and G. Tryggvason, "A DNS study of laminar bubbly flows in a vertical channel," *International Journal of Multiphase Flow*, vol. 32, no. 6, pp. 643 – 660, 2006.

[95] J. Lu and G. Tryggvason, "Effect of bubble deformability in turbulent bubbly upflow in a vertical channel," *Physics of Fluids (1994-present)*, vol. 20, no. 4, pp. –, 2008.

[96] 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.

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

[98] R. Clift, J. R. Grace, and M. E. Weber, *Bubbles, Drops, and Particles*. Academic Press, Inc., 1978.

[99] K. B. Deshpande and W. B. Zimmerman, "Simulation of interfacial mass transfer by droplet dynamics using the level set method," *Chemical Engineering Science*, vol. 61, pp. 6486–6498, 2006.

[100] V. Klimenko, "A generalized correlation for two-phase forced flow heat transfer," *International Journal of Heat and Mass Transfer*, vol. 31, no. 3, pp. 541 – 552, 1988.

[101] H. Kim and J. Buongiorno, "Detection of liquid-vapor-solid triple contact line in two-phase heat transfer phenomena using high-speed infrared thermometry," *International Journal of Multiphase Flow*, vol. 37, no. 2, pp. 166 – 172, 2011.

[102] A. Guion, D. Langewisch, and J. Buongiorno, "On the liquid microlayer underneath a vapor bubble growing at a heated wall," *8th International Conference on Multiphase Flow*, Jeju, Korea, May 26-31, 2013.

[103] B.-J. Yun, A. Splawski, S. Lo, and C.-H. Song, "Prediction of a subcooled boiling flow with advanced two-phase flow models," *Nuclear Engineering and Design*, no. 0, pp. –, 2011.

[104] J. O. Hinze, *Turbulence*. McGraw-Hill, New York, 1975.

[105] R. Mei and J. F. Klausner, "Shear lift force on spherical bubbles," *International Journal of Heat and Fluid Flow*, vol. 15, pp. 62–65, 1994.

[106] T. R. Auton, "The lift force on a spherical body in a rotational flow," *Journal of Fluid Mechanics*, vol. 183, pp. 199–218, 1987.

[107] B. B. Mikic, W. M. Rohsenow, and P. Griffith, "On bubble growth rates," *International Journal of Heat and Mass Transfer*, vol. 13, pp. 657–666, 1970.

[108] H. Ivey, "Relationships between bubble frequency, departure diameter and rise velocity in nucleate boiling," *International Journal of Heat and Mass Transfer*, vol. 10, no. 8, pp. 1023 – 1040, 1967.

[109] K. Stephan, *Heat Transfer in Condensation and Boiling*. Springer-Verlag, New York, 1992.

[110] M. Jakob, *Heat Transfer I*. Wiley and Sons, New York, 1949.

[111] H. Chi-Yeh and P. Griffith, "The mechanism of heat transfer in nucleate pool boiling-part i: Bubble initiaton, growth and departure," *International Journal of Heat and Mass Transfer*, vol. 8, no. 6, pp. 887 – 904, 1965.

[112] "Journal of vacuum science & technology: Part A-vacuums, surfaces & films," vol. 19, 2001.

[113] U. S. of Commerce on behalf of the United States of America, "National institute of standards and technology," 2011.

[114] A. Hasan and R. P. Roy, "Experiments on subcooled flow boiling heat transfer in a vertical annular channel," *International Journal of Heat and Mass Transfer*, vol. 33, pp. 2285–2293, 1990.

[115] R. P. Roy, S. Kang, J. A. Zarate, and A. Laporta, "Turbulent subcooled boiling flow - experiments and simulations," *Journal of Heat Transfer*, vol. 124, pp. 73–93, 2002.

[116] W. M. Rosenhow and J. A. Clark, "Heat transfer and pressure drop data for high heat flux densities to water at subcritical pressures," tech. rep., Massachusetts Institute of Technology for The Office of Naval Research, April 1, 1951.

[117] V. Tolubinsky and J. Ostrovsky, "On the mechanism of boiling heat transfer (vapour bubbles growth rate in the process of boiling of liquids, solutions, and binary mixtures)," *International Journal of Heat and Mass Transfer*, vol. 9, no. 12, pp. 1463 – 1470, 1966.

[118] R. Siegel and E. G. Keshock, "Effects of reduced gravity on nucleate boiling bubble dynamics in saturated water," *American Institute of Chemical Engineers Journal*, vol. 10, pp. 509–517, 1964.

[119] A. Faghri and Y. Zhang, *Transport Phenomena in Multiphase Systems*. Elsevier Science, 2006.

[120] D. R. Lide, ed., *CRC Handbook of Chemistry and Physics, 87th Ed.* CRC Press, 2006-2007.

[121] Y. Y. Jiang, H. Osada, M. Inagaki, and N. Horinouchi, "Dynamic modeling on bubble growth, detachment and heat transfer for hybrid-scheme ccomputation of nucleate boiling," *International Journal of Heat and Mass Transfer*, vol. 56, pp. 640–652, 2013.

Appendix A User Code Files

The boiling model is implemented in STAR-CCM+ by linking user libraries compiled from subroutines written in C as described in Section. The files (uclib.h, uclib.c, ucmodels.h, and ucmodels.h) used for building each coefficient used in the linearized boiling formulation are included below. Also included are the files used to build the force balance bubble departure model as described in section. The uclib.h file is the same for every user code library built, so it is only shown in Liquid Coefficient A files.

Liquid Coefficient A

```
/*
  ____________________________________________________________________________

 * 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()
{

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

 * userLiquidCoefficientA
 *****************************************************************************/

  ucfunc(
    /* void *func     */ userLiquid_A,
    /* char *type     */ UCFUNC_TYPE_SCALAR_BP,
    /* char *name     */ UCFUNC_TAG_HEATFLUX_LIQUID_A
  );

ucarg (
    /* void *func     */ userLiquid_A,
    /* char *type     */ "Face",
    /* char *name     */ "FaceCellIndex",
    /* int  size      */ sizeof(int[2])
  );

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

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

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

ucarg (
```

```
    /* void *func     */ userLiquid_A,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_BUBBLE_DIAMETER,
    /* int  size      */ sizeof(CoordReal)
  );

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

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


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


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


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

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

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

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

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

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

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

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

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

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

ucarg (
    /* void *func     */ userLiquid_A,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_LIQUID_HTC,
    /* 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_HEATFLUX_LIQUID_A "LiquidCoeff_A"

/* input field solver names */
#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_INFLUENCE_AREA "
    $BubbleInfluenceWallAreaFractionPhaseInteraction"
#define UCARG_TAG_WALL_DRY_AREA "$WallDryoutAreaFractionPhaseInteraction"
#define UCARG_TAG_W_VELOCITY "Phase0::W_Velocity"
#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 "$SurfaceTensionPhaseInteraction"
#define UCARG_TAG_BUBBLE_FREQUENCY "$BubbleDepartureFrequencyPhaseInteraction"
#define UCARG_TAG_LIQUID_HTC "$InternalWallHeatFluxCoefficientBLiquid"

/* function prototypes */
void userLiquid_A(
  Real*          result,
  int            size,
  int            (*fc)[2],
  Real*          DensityLiquid,
  Real*          DensityVapor,
  CoordReal*     NucleationSiteNumberDensityPhaseInteraction,
  CoordReal*     BubbleDepartureDiameterPhaseInteraction,
  CoordReal*     TemperatureLiquidCell,
  CoordReal*     LatentHeatPhaseInteraction,
  CoordReal*     BubbleInfluenceWallAreaFractionPhaseInteraction,
  CoordReal*     WallDryoutAreaFractionPhaseInteraction,
  Real*          W_VelocityLiquid,
  CoordReal*     DynamicViscosityLiquid,
```

```
    CoordReal*      UstarLiquid,
    CoordReal*      TemperaturePhaseInteraction,
    CoordReal*      TemperatureLiquidWall,
    CoordReal*      SpecificHeatLiquid,
    CoordReal*      ThermalConductivityLiquid,
    CoordReal*      SurfaceTensionPhaseInteraction,
    CoordReal*      Frequency,
    CoordReal*      LiquidHTC

);

#endif
```

```
/*
   ____________________________________________________________________________

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

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

 * userHeatFluxLiquidA
 *****************************************************************************/

void userLiquid_A(
  Real*          result,
  int            size,
  int            (*fc)[2],
  Real*          DensityLiquid,
  Real*          DensityVapor,
  CoordReal*     NucleationSiteNumberDensityPhaseInteraction,
  CoordReal*     BubbleDepartureDiameterPhaseInteraction,
  CoordReal*     TemperatureLiquidCell,
  CoordReal*     LatentHeatPhaseInteraction,
  CoordReal*     BubbleInfluenceWallAreaFractionPhaseInteraction,
  CoordReal*     WallDryoutAreaFractionPhaseInteraction,
  Real*          W_VelocityLiquid,
  CoordReal*     DynamicViscosityLiquid,
  CoordReal*     UstarLiquid,
  CoordReal*     TemperaturePhaseInteraction,
  CoordReal*     TemperatureLiquidWall,
  CoordReal*     SpecificHeatLiquid,
  CoordReal*     ThermalConductivityLiquid,
  CoordReal*     SurfaceTensionPhaseInteraction,
  CoordReal*     Frequency,
  CoordReal*     LiquidHTC
) {
  /* [0] declarations */
// Defining Constants in the Bubble Departure Model //
double pi       =       3.1416;
double D_l_max;

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 //
double C_s   = 1;         // Variable constant in the model //

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 //
```

```
double delta_R = 5E-7;
double g = 9.81;
/* [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[fc[i][0]];
        double c_pl = SpecificHeatLiquid[i];
        double k_l = ThermalConductivityLiquid[i];

        double sigma = SurfaceTensionPhaseInteraction[i];
        double c_ph = 500;
        double rho_h = 8000;
        double f = Frequency[i];
        double v_l = W_VelocityLiquid[fc[i][0]]; // Velocity in the z-
           direction (direction of flow)
        double Theta = 1.57;
        double h_l = LiquidHTC[i];
        double R = D_b/2;
        double Delta_max = 2E-6; // the maximum height of the microlayer

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

        double Ja_sup, Ja_sat;
        double DeltaT_sup = T_wall-T_sat;
        double DeltaT_sat = 3;

        Ja_sat = rho_l*c_pl*(DeltaT_sat)/(rho_v*h_fg);
        Ja_sup = rho_l*c_pl*(DeltaT_sup)/(rho_v*h_fg);
        double q_q = 0;
        double q_mL = 1E-9;
        if (T_wall>T_sat)
        {
                double N_max;
                // Adjust Nucleation Site Density for maximum:
                double t_g = R*R*pi/(4*b*b*Ja_sat*Ja_sat*eta_l);
                if (t_g > 1/f)
```

```
                { t_g = 1/f; }
                double N_b = t_g*f*N;
                double P = 1-exp(-N_b*pi*D_b*D_b/4);
                N_max = (1-P)*N;
                N=N_max;
                /*-----------------------------------------------------------------------------

                ---------------------Microlayer Evaporation
                    ----------------------------
                -----------------------------------------------------------------------------*/

                double V_mL;
                V_mL = 2*pi*(R/2.0)*(R/2.0)*Delta_max/3.0;
                q_mL = V_mL*rho_l*h_fg*f*N;
                /*
                    ------------------------------------------------------------------

                ---------------------Quenching Term
                    ------------------------------------
                -----------------------------------------------------------------------------*/

                double V_q, DeltaT_h;
                DeltaT_h = 2; // The superheat that will be quenched on the
                    heater surface.
                V_q = 2*pi*R*R*R/3;
                q_q = rho_h*c_ph*DeltaT_h*V_q*f*N;
        }
        double q_tot = q_mL + q_q;
        result[i] = -q_tot;
}
}
```

Liquid Coefficient B

```
/*
   ____________________________________________________________________________

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

 */
#include "ucmodels.h"
void uclib()
{
/*****************************************************************************

 * user
 *****************************************************************************/

  ucfunc(
    /* void *func     */ userLiquid_B,
    /* char *type     */ UCFUNC_TYPE_SCALAR_BP,
    /* char *name     */ UCFUNC_TAG_HEATFLUX_LIQUID_B
  );
ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Face",
    /* char *name     */ "FaceCellIndex",
    /* int  size      */ sizeof(int[2])
  );
  ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_LIQUID_DENSITY,
    /* int   size     */ sizeof(Real)
  );
ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_VAPOR_DENSITY,
    /* int   size     */ sizeof(Real)
  );
ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_NUCLEATION_SITES,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_BUBBLE_DIAMETER,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
```

```
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_LIQUID_TEMPERATURE_CELL,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_LATENT_HEAT,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_INFLUENCE_AREA,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_WALL_DRY_AREA,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_W_VELOCITY,
    /* int   size     */ sizeof(Real)
  );
ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_DYNAMIC_VISCOSITY,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_U_TAU,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_SATURATION_TEMPERATURE,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_LIQUID_TEMPERATURE_WALL,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
```

```
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_SPECIFIC_HEAT,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_THERMAL_CONDUCTIVITY,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_SURFACE_TENSION,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_BUBBLE_FREQUENCY,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_LIQUID_HTC,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_B,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_PREVIOUS_LIQUID_B,
    /* 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_HEATFLUX_LIQUID_B "LiquidCoeff_B"

/* input field solver names */
#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 "$CellTemperature"
#define UCARG_TAG_LATENT_HEAT "$LatentHeatPhaseInteraction"
#define UCARG_TAG_INFLUENCE_AREA "
    $BubbleInfluenceWallAreaFractionPhaseInteraction"
#define UCARG_TAG_WALL_DRY_AREA "$WallDryoutAreaFractionPhaseInteraction"
#define UCARG_TAG_W_VELOCITY "Phase0::W_Velocity"
#define UCARG_TAG_DYNAMIC_VISCOSITY "$DynamicViscosityLiquid"
#define UCARG_TAG_U_TAU "$UstarLiquid"
#define UCARG_TAG_SATURATION_TEMPERATURE "$TemperaturePhaseInteraction"
#define UCARG_TAG_LIQUID_TEMPERATURE_WALL "$WallTemperature"
#define UCARG_TAG_SPECIFIC_HEAT "$SpecificHeatLiquid"
#define UCARG_TAG_THERMAL_CONDUCTIVITY "$ThermalConductivityLiquid"
#define UCARG_TAG_SURFACE_TENSION "$SurfaceTensionPhaseInteraction"
#define UCARG_TAG_BUBBLE_FREQUENCY "$BubbleDepartureFrequencyPhaseInteraction"
#define UCARG_TAG_LIQUID_HTC "$InternalWallHeatFluxCoefficientBLiquid"
#define UCARG_TAG_PREVIOUS_LIQUID_B "$LiquidBMonitor"

/* function prototypes */
void userLiquid_B(
  Real*          result,
  int            size,
  int            (*fc)[2],
  Real*          DensityLiquid,
  Real*          DensityVapor,
  CoordReal*     NucleationSiteNumberDensityPhaseInteraction,
  CoordReal*     BubbleDepartureDiameterPhaseInteraction,
  CoordReal*     TemperatureLiquidCell,
  CoordReal*     LatentHeatPhaseInteraction,
  CoordReal*     BubbleInfluenceWallAreaFractionPhaseInteraction,
  CoordReal*     WallDryoutAreaFractionPhaseInteraction,
  Real*          W_VelocityLiquid,
  CoordReal*     DynamicViscosityLiquid,
  CoordReal*     UstarLiquid,
```

```
  CoordReal*      TemperaturePhaseInteraction,
  CoordReal*      TemperatureLiquidWall,
  CoordReal*      SpecificHeatLiquid,
  CoordReal*      ThermalConductivityLiquid,
  CoordReal*      SurfaceTensionPhaseInteraction,
  CoordReal*      Frequency,
  CoordReal*      LiquidHTC,
  CoordReal*      PreviousLiquidB
);
#endif
```

```
/*
   ____________________________________________________________________________

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

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

 * userHeatFluxLiquidB
 *****************************************************************************/

void userLiquid_B(
  Real*          result,
  int            size,
  int            (*fc)[2],
  Real*          DensityLiquid,
  Real*          DensityVapor,
  CoordReal*     NucleationSiteNumberDensityPhaseInteraction,
  CoordReal*     BubbleDepartureDiameterPhaseInteraction,
  CoordReal*     TemperatureLiquidCell,
  CoordReal*     LatentHeatPhaseInteraction,
  CoordReal*     BubbleInfluenceWallAreaFractionPhaseInteraction,
  CoordReal*     WallDryoutAreaFractionPhaseInteraction,
  Real*          W_VelocityLiquid,
  CoordReal*     DynamicViscosityLiquid,
  CoordReal*     UstarLiquid,
  CoordReal*     TemperaturePhaseInteraction,
  CoordReal*     TemperatureLiquidWall,
  CoordReal*     SpecificHeatLiquid,
  CoordReal*     ThermalConductivityLiquid,
  CoordReal*     SurfaceTensionPhaseInteraction,
  CoordReal*     Frequency,
  CoordReal*     LiquidHTC,
  CoordReal*     PreviousLiquidB
) {
  /* [0] declarations */
double pi       =       3.1416;
double v_l_center = W_VelocityLiquid[channel_center];

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 //
```

```
double delta_R = 5E-8;
double g = 9.81;
double F_A = 2;
/* [1] record user model activation and run parameters */
int i;
for (i = 0; i != size; ++i)
        {
        double N = NucleationSiteNumberDensityPhaseInteraction[i];
        double B_old = PreviousLiquidB[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[fc[i][0]];
        double c_pl = SpecificHeatLiquid[i];
        double k_l = ThermalConductivityLiquid[i];

        double alpha = 91.2*(pi/180);  // in radians
        double beta = 26.3*(pi/180);            // in radians
        double sigma = SurfaceTensionPhaseInteraction[i];
        double c_ph = 500;
        double rho_h = 8000;
        double f = Frequency[i];
        double v_l = W_VelocityLiquid[fc[i][0]]; // Velocity in the z-
           direction
        double Theta = 1.57;
        double h_l = LiquidHTC[i];
        double K_dry = WallDryoutAreaFractionPhaseInteraction[i];
        double Kquench = BubbleInfluenceWallAreaFractionPhaseInteraction[i];
        double R = D_b/2;
        double Delta_max = 2E-6; // the maximum height of the microlayer
        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 Re_b;

        double Ja_sup, Ja_sat;
        double DeltaT_sup = T_wall-T_sat;
        double DeltaT_sat = 3;
        double DeltaT_sub = T_sat - T_in;
        double D_m, D_l, D_l2, h_fc_tot, h_sc;
        D_m = 0;
        D_l = 0;
        D_l2 = 0;
        h_fc_tot = 0;
        h_sc = 0;
```

```
        Ja_sup = rho_l*c_pl*(DeltaT_sup)/(rho_v*h_fg);
        Ja_sat = rho_l*c_pl*(DeltaT_sat)/(rho_v*h_fg);
        double t_star;
/* ------------------------------------------------------------------
--------------------Forced Convection-----------------------------------
------------------------------------------------------------------*/
double h_fc, h_fc_totarea;

h_fc = (1.0-K_dry)*h_l;
h_fc_tot = h_fc*(1.0-Kquench);
/* ------------------------------------------------------------------
--------------------Nucleation Site Density-----------------------------
------------------------------------------------------------------*/
if (T_wall>T_sat)
        {
        double N_max;
        double t_g = R*R*pi/(4*b*b*Ja_sat*Ja_sat*eta_l);
        if (t_g > 1/f)
                { t_g = 1/f; }

        double N_b = t_g*f*N;
        double P = 1-exp(-N_b*pi*D_b*D_b/4);
        N_max = (1-P)*N;
        N=N_max;
        }
/* ------------------------------------------------------------------
--------------------Bubble Departure Diameter Value---------------------
------------------------------------------------------------------*/
if (T_wall > T_sat)
        {
        for (R = 1E-8; 0.01; R = R + delta_R)
        {
        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 U_rel = v_l_center-U;
        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));
        Re_b = 2*U*R/nu_l;
/* ------------------------------------------------------------------
--------------------Surface Tension Force-------------------------------
------------------------------------------------------------------*/
int i_dw = 40;
double F_sx, F_sy;
double d_w = 2*R/i_dw;

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 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 ------------------------------------
-------------------------------------------------------------------------------*/
double F_du, F_dux, F_duy;
double R_t; // Bubble growth rate, first derivative, second derivative
double R_tt;
double R_ttt;
double t;

t = R*R*pi/(4*b*b*Ja_sat*Ja_sat*eta_l);

R_t = 2*b*Ja_sat*sqrt(eta_l*t/pi);
R_tt  = b*Ja_sat*sqrt(eta_l/(pi*t));
R_ttt = -b*Ja_sat*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);
/* ------------------------------------------------------------------------
--------------------------Summation of Forces -------------------------------
-------------------------------------------------------------------------------*/
double Sum_Fx, Sum_Fy, Sum_Slide;
```

```
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 x-forces is greater than zero  ==> the bubble slides!
        if (D_m == 0 && Sum_Fx > 1E-15)
           {D_m= 2*R;}
// If sum of y-forces is greater than zero  ==> the bubble lifts off!
       if(D_l == 0 && Sum_Fy> 1E-15 )
         { D_l = 2*R; }
        if (D_l > 0 && D_m > 0 )
        {        break;}
}
        //Maximum Eo is set at 0.1:
        double D_l_max = sqrt(0.1*sigma/((rho_l-rho_v)*g));
        D_l2 = D_l_max;
        double R_l_max = D_l_max/2.0;
        delta_R = 5E-7;
/* ------------------------------------------------------------------------
--------------------Bubble Lift-off Diameter Value--------------------
------------------------------------------------------------------------*/
        for (R = 1E-7; R <= R_l_max; R = R + delta_R)
        {
        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 U_rel = v_l_center-U; // Relative velocity for once the bubble
           starts to slide
        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));
        Re_b = 2*U*R/nu_l;
        double Re_b_sliding = 2*U_rel*R/nu_l;
        /* ------------------------------------------------------------------
        --------------------Buoyancy Force ------------------------------
        ------------------------------------------------------------------*/
        double F_b = 4*pi*(R*R*R)*(rho_l - rho_v)*g/3;
        /* ------------------------------------------------------------------
        --------------------Unsteady Force ------------------------------
        ------------------------------------------------------------------*/
        double t_sliding, R_t_sliding, R_tt_sliding, R_ttt_sliding,
           F_du_sliding;
        double t_sliding = R*R*pi/(4*b*b*Ja_sup*Ja_sup*eta_l);
        double R_t_sliding = 2*b*Ja_sup*sqrt(eta_l*t_sliding/pi);
        double R_tt_sliding  = b*Ja_sup*sqrt(eta_l/(pi*t_sliding)); // include
            this for Yun's subcooling: - b*q_l/(s*h_fg*rho_v);
        double R_ttt_sliding = -b*Ja_sup*sqrt(eta_l/pi)*(pow(t_sliding,(-1.5))
           )/2;

        double F_du_sliding = -rho_l*pi*pow(R_t_sliding,2)*(R_t_sliding*
           R_ttt_sliding + 1.5*C_s*pow(R_tt_sliding,2));
        /* ------------------------------------------------------------------
```

```
        ----------------------Shear Lift Force ----------------------------------
        ------------------------------------------------------------------------*/
        double G_s_sliding = fabs(dU_dY*R/U_rel);
        double part_1_slide = 0.5*rho_l*U_rel*U_rel*pi*(R*R);
        double inpow_slide = (1/(Re_b_sliding*Re_b_sliding) + 0.014*
            G_s_sliding*G_s_sliding);
        double F_sL_sliding = part_1_slide*3.877*pow(G_s_sliding,0.5)*pow(
            inpow_slide,0.25);

        double Sum_Slide = F_du_sliding-F_b*cos(Theta)+F_sL_sliding;
        if (D_l2 == 0 && Sum_Slide > 1E-10)
        {D_l2 = 2*R;
        break;}
}
R = D_b/2;
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));
Re_b = 2*U*R/nu_l;
}
R = D_b/2;
/* ------------------------------------------------------------------------
----------------------Sliding Conduction ----------------------------------
------------------------------------------------------------------------*/
/*Sliding bubble effect is calculated using the properties of the fluid to
    reform the fluid
layer over which the bubble slides. Need to know both the bubble departure
    diameter (D_m) and
the bubble departure liftoff diameter (D_l). These values come from the force
    balance bubble
departure model. The sliding time is calculated based on a correlation by Basu
    .*/
double l, l2;
double q_sc = 5E-5;
double s, t_sl, t_g;
double a_sl, D_avg, R_f, A_sl;
double h_fc1, h_fc2;

if (D_l >= D_m )
        {
        if (T_wall - T_sat >0)
                {
                if (D_l2 > D_m)
                        {
s = 1/sqrt(N); // Spacing between bubbles
t_sl = s/v_l;
t_g = R*R*pi/(4*b*b*Ja_sup*Ja_sup*eta_l);

double fraction, D_sl, D_star, l, D_add;
fraction = t_sl*eta_l*Ja_sup/(15*(0.015+0.0023*sqrt(Re_b))*(0.04+0.023*sqrt(
    Ja_sub)));
D_sl = sqrt(fraction + D_m*D_m); // the bubble diameter after sliding a
    distance "s"
```

```
// Now check if the bubble is greater than D_l. If not, absorb a bubble and
   check again.
double N_merg, t_sl_star, s_star;

if (D_sl > D_l2)
        // Then need to calculate at what distance it reached D_l
        { t_sl_star = (D_l2*D_l2 - D_m*D_m)*(15*(0.015+0.0023*sqrt(Re_b))
           *(0.04+0.023*sqrt(Ja_sub)))/(eta_l*Ja_sup);
        s_star = t_sl_star* v_l;
        // And then calculate total sliding distance:
        l = s_star;
        }
// After sliding a distance "s":
if (D_sl < D_l2)
        {
        // loop (while bubble is sliding until it reaches D_l)
        for ( N_merg = 1; N_merg <= 1000; N_merg = N_merg + 1 )
                {       // Absorb a bubble
                        D_add = D_m;
                        D_star = pow(D_sl*D_sl*D_sl+D_add*D_add*D_add,0.3333);

                        // Check if it is now greater than D_l
                        if (D_star > D_l2)
                                { s_star = 0;
                                break; }

                        // Otherwise it slides again
                        D_sl = sqrt(fraction + D_star*D_star);

                        if (D_sl > D_l2)
                                // Then need to calculate at what distance it
                                   reached D_l
                                { t_sl_star = (D_l2*D_l2 - D_star*D_star)
                                   *(15*(0.015+0.0023*sqrt(Re_b))
                                   *(0.04+0.023*sqrt(Ja_sub)))/(eta_l*Ja_sup)
                                   ;
                                  s_star = t_sl_star* v_l;
                                  break;
                                }
                }
        // Then calculate total distance traveled by a bubble:
        l = s*N_merg + s_star;
        }
}
// If it predicts the lift-off diameter less than D_m then have it slide one
   bubble diameter in length
if(D_l2 <= D_m)
        {l2=D_m;}

// To calculate the total area swept by sliding bubbles:
D_avg = D_m + 0.5*(D_l2-D_m);   // average bubble size while sliding
a_sl = D_avg*l; // area swept by one bubble
s = 1/sqrt(N); // Spacing between bubbles
R_f = 1/(l/s+1);
```

```
// If it slides less than the spacing between two nucleation sites:
if (l<s)
        {R_f = 1;}
// Decrease the number of nucleation sites by using this reduction factor
N = R_f*N;
// The sliding conduction takes place in parts of the area until the thermal
   layer is reformed. This is calculated by equating the forced convection
   term
// to the transient conduction term to see when the forced convection term
   becomes a greater value.
t_star = k_l*k_l/(h_fc*h_fc*pi*eta_l);
if (t_star > (1/f) )
        { t_star = (1/f); }

// Calculate the heat flux for sliding conduction using the time for sliding
   conduction to occur:
h_sc = (2*k_l*N*f*a_sl*t_star/sqrt(pi*eta_l*t_star));
/* ------------------------------------------------------------------
-------------------Modify Forced Convection-----------------------------
------------------------------------------------------------------------*/
// Now to account for the area over which this occurs:
h_fc1 = h_fc*(1.0-Kquench)*(1.0-a_sl*N); // over heater unaffected by bubbles
if (h_fc1<0)
        {h_fc1 = 0;}

h_fc2 = h_fc*(1.0-Kquench)*a_sl*N*(1.0-t_star*f); // over the sliding
   convection area
h_fc_tot = h_fc1 + h_fc2;
                }
        }
//////////////////// If there is no sliding conduction because D_l <= D_m then:
if (T_wall > T_sat)
        {
        if (D_l < D_m)
                {
                l=D_m;
                D_avg = D_m;    // average bubble size while sliding
                a_sl = D_avg*l;
                t_star = k_l*k_l/(h_fc*h_fc*pi*eta_l);
                if (t_star > (1/f) )
                        { t_star = (1/f); }
                // Calculate the heat flux for sliding conduction using the
                   time for sliding conduction to occur:
                h_sc = (2*k_l*N*f*a_sl*t_star/sqrt(pi*eta_l*t_star));
                }
        }
double h_tot = (h_fc_tot + h_sc);
result[i] = 0.1*(h_tot-B_old)+B_old;
        }
}
```

Liquid Coefficient C

```
/*
   ____________________________________________________________________________

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

 */
#include "ucmodels.h"
void uclib()
{
/*****************************************************************************

 * user
 *****************************************************************************/

  ucfunc(
    /* void *func     */ userLiquid_C,
    /* char *type     */ UCFUNC_TYPE_SCALAR_BP,
    /* char *name     */ UCFUNC_TAG_HEATFLUX_LIQUID_C
  );
ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Face",
    /* char *name     */ "FaceCellIndex",
    /* int  size      */ sizeof(int[2])
  );
  ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_LIQUID_DENSITY,
    /* int   size     */ sizeof(Real)
  );
ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_VAPOR_DENSITY,
    /* int   size     */ sizeof(Real)
  );
ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_NUCLEATION_SITES,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_BUBBLE_DIAMETER,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
```

```
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_LIQUID_TEMPERATURE_CELL,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_LATENT_HEAT,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_INFLUENCE_AREA,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_WALL_DRY_AREA,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_W_VELOCITY,
    /* int  size      */ sizeof(Real)
  );
ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_DYNAMIC_VISCOSITY,
    /* int  size      */ sizeof(CoordReal)
  );

ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_U_TAU,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_SATURATION_TEMPERATURE,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_LIQUID_TEMPERATURE_WALL,
    /* int  size      */ sizeof(CoordReal)
  );
```

```
ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_SPECIFIC_HEAT,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_THERMAL_CONDUCTIVITY,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_SURFACE_TENSION,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_BUBBLE_FREQUENCY,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_LIQUID_HTC,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userLiquid_C,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_PREVIOUS_LIQUID_C,
    /* 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_HEATFLUX_LIQUID_C "LiquidCoeff_C"
/* input field solver names */
#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_INFLUENCE_AREA "
    $BubbleInfluenceWallAreaFractionPhaseInteraction"
#define UCARG_TAG_WALL_DRY_AREA "$WallDryoutAreaFractionPhaseInteraction"
#define UCARG_TAG_W_VELOCITY "Phase0::W_Velocity"
#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 "$SurfaceTensionPhaseInteraction"
#define UCARG_TAG_BUBBLE_FREQUENCY "$BubbleDepartureFrequencyPhaseInteraction"
#define UCARG_TAG_LIQUID_HTC "$InternalWallHeatFluxCoefficientBLiquid"
#define UCARG_TAG_PREVIOUS_LIQUID_C "$LiquidCMonitor"

/* function prototypes */
void userLiquid_C(
  Real*          result,
  int            size,
  int            (*fc)[2],
  Real*          DensityLiquid,
  Real*          DensityVapor,
  CoordReal*     NucleationSiteNumberDensityPhaseInteraction,
  CoordReal*     BubbleDepartureDiameterPhaseInteraction,
  CoordReal*     TemperatureLiquidCell,
  CoordReal*     LatentHeatPhaseInteraction,
  CoordReal*     BubbleInfluenceWallAreaFractionPhaseInteraction,
  CoordReal*     WallDryoutAreaFractionPhaseInteraction,
  Real*          W_VelocityLiquid,
  CoordReal*     DynamicViscosityLiquid,
  CoordReal*     UstarLiquid,
  CoordReal*     TemperaturePhaseInteraction,
  CoordReal*     TemperatureLiquidWall,
```

```
  CoordReal*      SpecificHeatLiquid,
  CoordReal*      ThermalConductivityLiquid,
  CoordReal*      SurfaceTensionPhaseInteraction,
  CoordReal*      Frequency,
  CoordReal*      LiquidHTC,
  CoordReal*      PreviousLiquidC
);
#endif
```

```
/*
   ____________________________________________________________________________

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

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

 * userHeatFluxLiquidC
 *****************************************************************************/

void userLiquid_C(
  Real*          result,
  int            size,
  int            (*fc)[2],
  Real*          DensityLiquid,
  Real*          DensityVapor,
  CoordReal*     NucleationSiteNumberDensityPhaseInteraction,
  CoordReal*     BubbleDepartureDiameterPhaseInteraction,
  CoordReal*     TemperatureLiquidCell,
  CoordReal*     LatentHeatPhaseInteraction,
  CoordReal*     BubbleInfluenceWallAreaFractionPhaseInteraction,
  CoordReal*     WallDryoutAreaFractionPhaseInteraction,
  Real*                  W_VelocityLiquid,
  CoordReal*     DynamicViscosityLiquid,
  CoordReal*     UstarLiquid,
  CoordReal*     TemperaturePhaseInteraction,
  CoordReal*     TemperatureLiquidWall,
  CoordReal*     SpecificHeatLiquid,
  CoordReal*     ThermalConductivityLiquid,
  CoordReal*     SurfaceTensionPhaseInteraction,
  CoordReal*     Frequency,
  CoordReal*     LiquidHTC,
  CoordReal*     PreviousLiquidC
) {
  /* [0] declarations */
double pi       =       3.1416;
double v_l_center = W_VelocityLiquid[channel_center];

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 //
```

```
double delta_R = 5E-8;
double g = 9.81;
double F_A = 2;

/* [1] record user model activation and run parameters */
int i;
for (i = 0; i != size; ++i)
        {
        double N = NucleationSiteNumberDensityPhaseInteraction[i];
        double C_old = PreviousLiquidC[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[fc[i][0]];
        double c_pl = SpecificHeatLiquid[i];
        double k_l = ThermalConductivityLiquid[i];

        double alpha = 91.2*(pi/180);  // in radians
        double beta = 26.3*(pi/180);            // in radians
        double sigma = SurfaceTensionPhaseInteraction[i];
        double c_ph = 500;
        double rho_h = 8000;
        double f = Frequency[i];
        double v_l = W_VelocityLiquid[fc[i][0]]; // Velocity in the z-
           direction
        double Theta = 1.57;
        double h_l = LiquidHTC[i];
        double K_dry = WallDryoutAreaFractionPhaseInteraction[i];
        double Kquench = BubbleInfluenceWallAreaFractionPhaseInteraction[i];
        double R = D_b/2;
        double Delta_max = 2E-6; // the maximum height of the microlayer
        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 Re_b;

        double Ja_sup, Ja_sat;
        double DeltaT_sup = T_wall-T_sat;
        double DeltaT_sat = 3;
        double D_m, D_l, D_l2, h_fc_tot, h_sc;

        D_m = 0;
        D_l = 0;
        D_l2 = 0;
        h_fc_tot = 0;
        h_sc = 0;
```

```
        Ja_sup = rho_l*c_pl*(DeltaT_sup)/(rho_v*h_fg);
        Ja_sat = rho_l*c_pl*(DeltaT_sat)/(rho_v*h_fg);
        double t_star;
/* ------------------------------------------------------------------
--------------------Forced Convection-----------------------------------
------------------------------------------------------------------*/
double h_fc, h_fc_totarea;

h_fc = (1.0-K_dry)*h_l;
h_fc_tot = h_fc*(1.0-Kquench);
/* ------------------------------------------------------------------
--------------------Nucleation Site Density-----------------------------
------------------------------------------------------------------*/
if (T_wall>T_sat)
        {
        double N_max;
        double t_g = R*R*pi/(4*b*b*Ja_sat*Ja_sat*eta_l);
        if (t_g > 1/f)
                { t_g = 1/f; }

        double N_b = t_g*f*N;
        double P = 1-exp(-N_b*pi*D_b*D_b/4);
        N_max = (1-P)*N;
        N=N_max;
        }
/* ------------------------------------------------------------------
--------------------Bubble Departure Diameter Value---------------------
------------------------------------------------------------------*/
if (T_wall > T_sat)
        {
        for (R = 1E-8; 0.01; R = R + delta_R)
        {
    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 U_rel = v_l_center-U;
        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));
        Re_b = 2*U*R/nu_l;

/* ------------------------------------------------------------------
--------------------Surface Tension Force-------------------------------
------------------------------------------------------------------*/
int i_dw = 40;
double F_sx, F_sy;
double d_w = 2*R/i_dw;

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 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 ———————————————————————————
————————————————————————————————————————————————————————————*/
double F_du, F_dux, F_duy;

double R_t; // Bubble growth rate, first derivative, second derivative
double R_tt;
double R_ttt;
double t;
t = R*R*pi/(4*b*b*Ja_sat*Ja_sat*eta_l);

R_t = 2*b*Ja_sat*sqrt(eta_l*t/pi);
R_tt  = b*Ja_sat*sqrt(eta_l/(pi*t));
R_ttt = -b*Ja_sat*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);
```

```
/* ------------------------------------------------------------------------
---------------------Summation of Forces ---------------------------------
------------------------------------------------------------------------*/
double Sum_Fx, Sum_Fy, Sum_Slide;

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 x-forces is greater than zero  ==> the bubble slides!
        if (D_m == 0 && Sum_Fx > 1E-15)
           {D_m= 2*R;}

// If sum of y-forces is greater than zero  ==> the bubble lifts off!
       if(D_l == 0 && Sum_Fy> 1E-15 )
         { D_l = 2*R; }

        if (D_l > 0 && D_m > 0 )
        {        break;}
}
        //Maximum Eo is set at 0.1:
        double D_l_max = sqrt(0.1*sigma/((rho_l-rho_v)*g));
        D_l2 = D_l_max;
        double R_l_max = D_l_max/2.0;
/* ------------------------------------------------------------------------
---------------------Bubble Lift-off Diameter Value----------------------
------------------------------------------------------------------------*/
        for (R = 1E-7; R <= R_l_max; R = R + delta_R)
        {
        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 U_rel = v_l_center-U; // Relative velocity for once the bubble
           starts to slide
        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));
        Re_b = 2*U*R/nu_l;
        double Re_b_sliding = 2*U_rel*R/nu_l;
        /* ----------------------------------------------------------------
        ---------------------Buoyancy Force ------------------------------
        ----------------------------------------------------------------*/
        double F_b = 4*pi*(R*R*R)*(rho_l - rho_v)*g/3;
        /* ----------------------------------------------------------------
        ---------------------Unsteady Force ------------------------------
        ----------------------------------------------------------------*/
        double t_sliding, R_t_sliding, R_tt_sliding, R_ttt_sliding,
           F_du_sliding;
        double t_sliding = R*R*pi/(4*b*b*Ja_sup*Ja_sup*eta_l);
        double R_t_sliding = 2*b*Ja_sup*sqrt(eta_l*t_sliding/pi);
        double R_tt_sliding  = b*Ja_sup*sqrt(eta_l/(pi*t_sliding)); // include
            this for Yun's subcooling: - b*q_l/(s*h_fg*rho_v);
        double R_ttt_sliding = -b*Ja_sup*sqrt(eta_l/pi)*(pow(t_sliding,(-1.5))
           )/2;
```

```
        double F_du_sliding = -rho_l*pi*pow(R_t_sliding,2)*(R_t_sliding*
            R_ttt_sliding + 1.5*C_s*pow(R_tt_sliding,2));

        /* ------------------------------------------------------------
        --------------------Shear Lift Force ----------------------------
        ------------------------------------------------------------*/
        double G_s_sliding = fabs(dU_dY*R/U_rel);
        double part_1_slide = 0.5*rho_l*U_rel*U_rel*pi*(R*R);
        double inpow_slide = (1/(Re_b_sliding*Re_b_sliding) + 0.014*
            G_s_sliding*G_s_sliding);
        double F_sL_sliding = part_1_slide*3.877*pow(G_s_sliding,0.5)*pow(
            inpow_slide,0.25);

        double Sum_Slide = F_du_sliding-F_b*cos(Theta)+F_sL_sliding;

        if (D_l2 == 0 && Sum_Slide > 1E-10)
        {D_l2 = 2*R;
        break;}
}

R = D_b/2;
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));
Re_b = 2*U*R/nu_l;
}
R = D_b/2;
/* ------------------------------------------------------------
--------------------Sliding Conduction ----------------------------
------------------------------------------------------------*/
/*Sliding bubble effect is calculated using the properties of the fluid to
    reform the fluid
layer over which the bubble slides. Need to know both the bubble departure
    diameter (D_m) and
the bubble departure liftoff diameter (D_l). These values come from the force
    balance bubble
departure model. The sliding time is calculated based on a correlation by Basu
    .*/
double l, l2;
double q_sc = 5E-5;
double s, t_sl, t_g;
double a_sl, D_avg, R_f, A_sl;
double h_fc1, h_fc2;

if (D_l >= D_m )
        {
        if (T_wall - T_sat >0)
                {
                if (D_l2 > D_m)
                        {
s = 1/sqrt(N); // Spacing between bubbles
t_sl = s/v_l;
t_g = R*R*pi/(4*b*b*Ja_sup*Ja_sup*eta_l);
```

```
double fraction, D_sl, D_star, l, D_add;

fraction = t_sl*eta_l*Ja_sup/(15*(0.015+0.0023*sqrt(Re_b))*(0.04+0.023*sqrt(
    Ja_sub)));
D_sl = sqrt(fraction + D_m*D_m); // the bubble diameter after sliding a
    distance "s"

// Now check if the bubble is greater than D_l. If not, absorb a bubble and
    check again.
double N_merg, t_sl_star, s_star;

if (D_sl > D_l2)
        // Then need to calculate at what distance it reached D_l
        { t_sl_star = (D_l2*D_l2 - D_m*D_m)*(15*(0.015+0.0023*sqrt(Re_b))
            *(0.04+0.023*sqrt(Ja_sub)))/(eta_l*Ja_sup);
        s_star = t_sl_star* v_l;
        // And then calculate total sliding distance:
        l = s_star;
        }

// After sliding a distance "s":
if (D_sl < D_l2)
        {
        // loop (while bubble is sliding until it reaches D_l)
        for ( N_merg = 1; N_merg <= 1000; N_merg = N_merg + 1 )
                {       // Absorb a bubble
                        D_add = D_m;
                        D_star = pow(D_sl*D_sl*D_sl+D_add*D_add*D_add,0.3333);

                        // Check if it is now greater than D_l
                        if (D_star > D_l2)
                                { s_star = 0;
                                break; }

                        // Otherwise it slides again
                        D_sl = sqrt(fraction + D_star*D_star);

                        if (D_sl > D_l2)
                                // Then need to calculate at what distance it
                                    reached D_l
                                { t_sl_star = (D_l2*D_l2 - D_star*D_star)
                                    *(15*(0.015+0.0023*sqrt(Re_b))
                                    *(0.04+0.023*sqrt(Ja_sub)))/(eta_l*Ja_sup)
                                    ;
                                  s_star = t_sl_star* v_l;
                                  break;
                                }
                }
        // Then calculate total distance traveled by a bubble:
        l = s*N_merg + s_star;
        }
        l2 = l;
}
```

```
// If it predicts the lift-off diameter less than D_m then have it slide one
   bubble diameter in length
if(D_l2 <= D_m)
        {l2=D_m;}
// To calculate the total area swept by sliding bubbles:
D_avg = D_m + 0.5*(D_l2-D_m);   // average bubble size while sliding
a_sl = D_avg*l2; // area swept by one bubble
s = 1/sqrt(N); // Spacing between bubbles
R_f = 1/(l2/s+1);

// If it slides less than the spacing between two nucleation sites:
if (l2<s)
        {R_f = 1;}
// Decrease the number of nucleation sites by using this reduction factor
N = R_f*N;
// The sliding conduction takes place in parts of the area until the thermal
   layer is reformed. This is calculated by equating the forced convection
   term
// to the transient conduction term to see when the forced convection term
   becomes a greater value.
t_star = k_l*k_l/(h_fc*h_fc*pi*eta_l);
if (t_star > (1/f) )
        { t_star = (1/f); }
// Calculate the heat flux for sliding conduction using the time for sliding
   conduction to occur:
h_sc = (2*k_l*N*f*a_sl*t_star/sqrt(pi*eta_l*t_star));

/* ------------------------------------------------------------------
-------------------Modify Forced Convection-----------------------------
------------------------------------------------------------------*/
// Now to account for the area over which this occurs:
h_fc1 = h_fc*(1.0-Kquench)*(1.0-a_sl*N); // over heater unaffected by bubbles
if (h_fc1<0)
        {h_fc1 = 0;}

h_fc2 = h_fc*(1.0-Kquench)*a_sl*N*(1.0-t_star*f); // over the sliding
   convection area
h_fc_tot = h_fc1 + h_fc2;
                }
        }
///////////////// If there is no sliding conduction because D_l <= D_m then:
if (T_wall > T_sat)
        {
        if (D_l < D_m)
                {
                l2=D_m;
                D_avg = D_m;    // average bubble size while sliding
                a_sl = D_avg*l2;
                t_star = k_l*k_l/(h_fc*h_fc*pi*eta_l);
                if (t_star > (1/f) )
                        { t_star = (1/f); }
                // Calculate the heat flux for sliding conduction using the
                   time for sliding conduction to occur:
                h_sc = (2*k_l*N*f*a_sl*t_star/sqrt(pi*eta_l*t_star));
```

```
                }
        }
double h_tot = (h_fc_tot + h_sc);
result[i] = 0.1*(-h_tot-C_old)+C_old;

        }

}
```

Phase Interaction Coefficient A

```
/*
   ____________________________________________________________________________

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

 */
#include "ucmodels.h"
void uclib()
{
/*****************************************************************************

 * user
 *****************************************************************************/

  ucfunc(
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ UCFUNC_TYPE_SCALAR_BP,
    /* char *name     */ UCFUNC_TAG_HEATFLUX_LIQUID_A
  );
ucarg (
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Face",
    /* char *name     */ "FaceCellIndex",
    /* int  size      */ sizeof(int[2])
  );
  ucarg (
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_LIQUID_DENSITY,
    /* int   size     */ sizeof(Real)
  );
ucarg (
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_VAPOR_DENSITY,
    /* int   size     */ sizeof(Real)
  );
ucarg (
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_NUCLEATION_SITES,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_BUBBLE_DIAMETER,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
```

```
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_LIQUID_TEMPERATURE_CELL,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_LATENT_HEAT,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_INFLUENCE_AREA,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_WALL_DRY_AREA,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_W_VELOCITY,
    /* int  size      */ sizeof(Real)
  );
ucarg (
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_DYNAMIC_VISCOSITY,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_U_TAU,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_SATURATION_TEMPERATURE,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_LIQUID_TEMPERATURE_WALL,
    /* int  size      */ sizeof(CoordReal)
  );
ucarg (
```

```
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_SPECIFIC_HEAT,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_THERMAL_CONDUCTIVITY,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Cell",
    /* char *variable */ UCARG_TAG_SURFACE_TENSION,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_BUBBLE_FREQUENCY,
    /* int   size     */ sizeof(CoordReal)
  );
ucarg (
    /* void *func     */ userPhaseInteraction_A,
    /* char *type     */ "Face",
    /* char *variable */ UCARG_TAG_LIQUID_HTC,
    /* 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_HEATFLUX_LIQUID_A "PhaseInteractionCoeff_A"

/* input field solver names */
#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_INFLUENCE_AREA "
    $BubbleInfluenceWallAreaFractionPhaseInteraction"
#define UCARG_TAG_WALL_DRY_AREA "$WallDryoutAreaFractionPhaseInteraction"
#define UCARG_TAG_W_VELOCITY "Phase0::W_Velocity"
#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 "$SurfaceTensionPhaseInteraction"
#define UCARG_TAG_BUBBLE_FREQUENCY "$BubbleDepartureFrequencyPhaseInteraction"
#define UCARG_TAG_LIQUID_HTC "$InternalWallHeatFluxCoefficientBLiquid"

/* function prototypes */
void userPhaseInteraction_A(
  Real*          result,
  int            size,
  int            (*fc)[2],
  Real*          DensityLiquid,
  Real*          DensityVapor,
  CoordReal*     NucleationSiteNumberDensityPhaseInteraction,
  CoordReal*     BubbleDepartureDiameterPhaseInteraction,
  CoordReal*     TemperatureLiquidCell,
  CoordReal*     LatentHeatPhaseInteraction,
  CoordReal*     BubbleInfluenceWallAreaFractionPhaseInteraction,
  CoordReal*     WallDryoutAreaFractionPhaseInteraction,
  Real*          W_VelocityLiquid,
  CoordReal*     DynamicViscosityLiquid,
  CoordReal*     UstarLiquid,
```

```
  CoordReal*      TemperaturePhaseInteraction,
  CoordReal*      TemperatureLiquidWall,
  CoordReal*      SpecificHeatLiquid,
  CoordReal*      ThermalConductivityLiquid,
  CoordReal*      SurfaceTensionPhaseInteraction,
  CoordReal*      Frequency,
  CoordReal*      LiquidHTC
);
#endif
```

```
/*
   ____________________________________________________________________________

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

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

 * userHeatFluxPhaseInteractionA
 *****************************************************************************/

void userPhaseInteraction_A(
  Real*          result,
  int            size,
  int            (*fc)[2],
  Real*          DensityLiquid,
  Real*          DensityVapor,
  CoordReal*     NucleationSiteNumberDensityPhaseInteraction,
  CoordReal*     BubbleDepartureDiameterPhaseInteraction,
  CoordReal*     TemperatureLiquidCell,
  CoordReal*     LatentHeatPhaseInteraction,
  CoordReal*     BubbleInfluenceWallAreaFractionPhaseInteraction,
  CoordReal*     WallDryoutAreaFractionPhaseInteraction,
  Real*          W_VelocityLiquid,
  CoordReal*     DynamicViscosityLiquid,
  CoordReal*     UstarLiquid,
  CoordReal*     TemperaturePhaseInteraction,
  CoordReal*     TemperatureLiquidWall,
  CoordReal*     SpecificHeatLiquid,
  CoordReal*     ThermalConductivityLiquid,
  CoordReal*     SurfaceTensionPhaseInteraction,
  CoordReal*     Frequency,
  CoordReal*     LiquidHTC
) {
double pi       =       3.1416;
// 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[fc[i][0]];
        double c_pl = SpecificHeatLiquid[i];
        double k_l = ThermalConductivityLiquid[i];

        double sigma = SurfaceTensionPhaseInteraction[i];
        double c_ph = 500;
        double rho_h = 8000;
        double f = Frequency[i];
        double v_l = W_VelocityLiquid[fc[i][0]]; // Velocity in the z-
           direction for cell
        double Theta = 1.57;
        double h_l = LiquidHTC[i];

        double R = D_b/2;
        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 Ja_sup, Ja_sat;  // Ja superheat and Ja subcooling
        double DeltaT_sup = T_wall-T_sat;
        double DeltaT_sat = 3;
        Ja_sat = rho_l*c_pl*(DeltaT_sat)/(rho_v*h_fg);

        // First need to check if nucleation site density is physical for the
           heater size or not. If not, then limit it
        if (T_wall>T_sat)
                {
                double N_max;

                // Adjust Nucleation Site Density for maximum:
                double t_g = R*R*pi/(4*b*b*Ja_sat*Ja_sat*eta_l);
                if (t_g > 1/f)
                        { t_g = 1/f; }

                double N_b = t_g*f*N;
                double size = pi*R*R;
                double P = 1-exp(-N_b*pi*D_b*D_b/4);

                N_max = (1-P)*N;
                N=N_max;
                }
/* ------------------------------------------------------------------
-------------------Initial Bubble Growth-----------------------------
------------------------------------------------------------------*/
                double q_e_i = N*f*pi*D_b*D_b*D_b*rho_v*h_fg/6;
                double q_tot = q_e_i;
                result[i] = -q_tot;
        }
}
```

Force Balance Bubble Departure Diameter

```
/*
   ____________________________________________________________________________

 * 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_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",
    /* char *variable */ UCARG_TAG_THERMAL_CONDUCTIVITY,
    /* int   size     */ sizeof(CoordReal)
  );

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

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

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

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



}
```

```
/*
 ____________________________________________________________________________

 * 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*     BoundaryHeatFluxLiquid,
  CoordReal*     Beta,
```

```
  CoordReal*      theta ,
  CoordReal*      ThermalConductivityLiquid ,
  CoordReal*      TemperatureLiquid ,
  CoordReal*      TemperaturePhaseInteraction ,
  CoordReal*      LatentHeatPhaseInteraction

) ;


#endif
```

```
/*
    ____________________________________________________________________________

 * ucfbmodels.c: implementation for shareable application-specific ucode
     models
 *
    ____________________________________________________________________________

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

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

void userBubbleDiameter(
  Real*          result,
  int            size,
  CoordReal*     SurfaceTensionPhaseInteraction,
  Real*          DensityLiquid,
  Real*          DensityVapor,
  CoordReal*     DynamicViscosityLiquid,
  CoordReal*     UstarLiquid,
  CoordReal*     Alpha,
  CoordReal*     SpecificHeatLiquid,
  CoordReal*     BoundaryHeatFluxLiquid,
  CoordReal*     Beta,
  CoordReal*     theta,
  CoordReal*     ThermalConductivityLiquid,
  CoordReal*     TemperatureLiquid,
  CoordReal*     TemperaturePhaseInteraction,
  CoordReal*     LatentHeatPhaseInteraction
) {
        // Defining Constants in the Bubble Departure Model //
        double pi       =       3.1416;
        double g =              9.81; // Gravitational Constant [m/s^2] //

        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 //

        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 //
        double C_s = 1;         // Variable constant in the model //

        /* [1] record user model activation and run parameters */
        double delta_R = 5E-8; // The change in the guess radius R for the
           loop [m]
        double R; // Bubble radius for departure [m]
        // First Loop through all the cells
```

```
        int i;
        for (i = 0; i != size; ++i)
                {
                double Theta = theta[i];
                double alpha = 100*(pi/180);  // in radians
                double beta = 25*(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);

                for (R = 1E-8; R <= 1E-1; R = R + delta_R)
                        {
                        double u_star = UstarLiquid[i];
                        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 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-----------------------------
------------------------------------------------------------------*/
int i_dw = 40;
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 ————————————————————————————————————
————————————————————————————————————————————————————————————————————————————*/
double Ja;
double T_sat = TemperaturePhaseInteraction[i];
double T_wall = TemperatureLiquid[i];
double DeltaT_sat = 3;
double F_du = 0;
double F_dux = 0;
double F_duy = 0;

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

double R_t; // Bubble growth rate, first derivative, second derivative
double R_tt;
double R_ttt;
double t;
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));
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);
/* ————————————————————————————————————————————————————————————————————————
————————————————————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 x-forces is greater than zero  ==> the bubble slides!
        if (Sum_Fx > 1E-15)
           {     result[i] = 2*R;
                 break;  }

// If sum of y-forces is greater than zero  ==> the bubble lifts off!
       if(Sum_Fy> 1E-15 )
         {       result[i] = 2*R;
                 break;    }
                 }
        }
}
```

Appendix B Field Function Code

The code employed in the calculation of the field functions used in the new wall boiling model is provided in this section.

Equivalent Sand-Grain Roughness

```
R_a = 5/3*3.14*$NucleationSiteNumberDensityPhaseInteraction*$t_g*
    $BubbleDepartureFrequencyPhaseInteraction*pow((
    $BubbleDepartureDiameterPhaseInteraction/2),3)

t_g = pow($BubbleDepartureDiameterPhaseInteraction/2,2)*3.14/(4*1.56*1.56*
    $Ja_sat*$Ja_sat*$ThermalConductivityLiquid/($DensityLiquid*
    $SpecificHeatLiquid))

Ja_sat = $DensityLiquid*$SpecificHeatLiquid*3/($DensityVapor*
    $LatentHeatPhaseInteraction)
```

Test Section Angle

```
Theta = acos(-dot($$WallNormal, $$gravity_direction))

WallNormal = $$Area / mag($$Area)

GravityDirection  = $$gravity / mag($$gravity)
```